LA-9246-M
Manual

UC-15
Issued: July 1982

LA--9246-M

DE82 021850

User's Manual for a Measurement Simulation Code

E. A. Kern

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

CONTENTS (cont)

TABLES

FIGURES

USER'S MANUAL FOR A MEASUREMENT SIMULATION CODE

by

E. A. Kern

ABSTRACT

The MEASIM code has been developed primarily for modeling process measurements in materials processing facilities associated with the nuclear fuel cycle. In addition, the code computes materials balances and the summation of materials balances along with associated variances. The code has been used primarily in performance assessment of materials' accounting systems. This report provides the necessary information for a potential user to employ the code in these applications. A number of examples that demonstrate most of the capabilities of the code are provided.

I. INTRODUCTION

The MEASIM (MEAsurement SIMulation) code was developed as one in a set of three codes to be used for diversion-sensitivity studies in nuclear materials processing facilities. MEASIM, as the name implies, simulates process measurements. The other two codes in this set are MODEL[1] and DECANAL.[2] MODEL is used to simulate process dynamics, and DECANAL (DECision ANALysis) uses measured values to determine the likelihood of missing material. MEASIM applies measurement errors to the true values of the process variables generated by MODEL. These measured values of process variables are the primary input to DECANAL.

In addition to simulating process measured values, MEASIM performs numerous other useful calculations. Input data to the

DECANAL code can become relatively complex for large process simulations. MEASIM writes the complete DECANAL input file. Other MEASIM calculations include total inventories, net transfers, materials balances, cumulative summations (CUSUMs) of materials balances, and measurement error variances associated with each of the above quantities. These calculations serve as a useful check on the input file to DECANAL because DECANAL computes the same quantities. MEASIM also has a built-in multirun capability wherein measurement calculations can be repeated with different random numbers. At the end of these multirun simulations, MEASIM computes the sample CUSUM variance and compares this sample with the theoretical or propagated CUSUM variance. This computation has proven to be a very powerful tool for checking measurement and variance calculations.

Except for the array sizes, MEASIM is completely process independent because all specific process information is contained in the input data file. This process independence is made possible through the modular manner in which MEASIM handles the process accounting areas. Array sizes are set in the code by a FORTRAN PARAMETER statement in which all the parameters are clearly defined.

Another useful feature of MEASIM is the capability to reduce the dimensions of subsequent DECANAL calculations. Generally, DECANAL treats each transfer and each inventory separately. For large processes, this approach can lead to long computation times and large array sizes for the DECANAL code. MEASIM provides the user with options of either combining the transfers into one equivalent transfer and the inventories into one equivalent inventory or only combining the inventories into one equivalent inventory. Reducing the dimensions of the problem in this manner can significantly reduce the DECANAL computation times and array sizes. However, for some problems these simplifications introduce errors into the variance and covariance calculations, so care must be taken in exercising these options.

The user can select either multiplicative, additive, or a combination of multiplicative and additive measurement error

models. Each measurement can have a maximum of one uncorrelated and two correlated error components. These errors are assumed to be normally distributed with mean zero and standard deviation provided by the user.

The free-format input data file contains the standard deviations of measurement errors along with the necessary inputs to define the process. This format results in a relatively simple input data file.

II. CODE STRUCTURE

A. Subroutine Summary

MEASIM is a highly modularized code with a relatively large number of subroutines. In most cases each subroutine performs only one type of calculation, so the code is relatively easy to understand and modify. A brief description of each of the MEASIM subroutines follows.

MAIN. This is the main driver for the MEASIM code. Files are opened and control is transferred to subroutine MESDRV. MAIN contains the logic for the Monte Carlo and multiple-run options available with the code. At the end of the simulation, MAIN transfers control to subroutine CLOSEM.

BLANKS. Reads blank lines of input data. These lines can be used for commenting purposes.

CLOSEM. Closes all the files and terminates the run.

COLUMN. Computes the measured inventory components for a pulsed column.

CUSUM. Computes the inventory and transfer variances and covariances, the *CUSUM*, and the *CUSUM* variance.

DRAND. Generates uniform random numbers in the interval 0-1.

INV1. Computes the measured inventory for a container where the inventory is equal to the measured value.

INV2. Computes the measured inventory for a container where the inventory is equal to the product of two measured values.

MASAGE. Computes variances and covariances for individual transfer and inventory components. Writes output file to DECANAL for the case where no reductions are made in the number of transfer and inventory components.

MASCUS. Combines the variances and covariances computed by subroutine MASAGE into a form suitable for the CUSUM variance calculation. Writes an output file to DECANAL for the case where all the inventories are combined and reduced to one inventory.

MEASR. Computes measured values for individual process variables.

MESDRV. Selects the set of desired transfer and inventory measurements for the process and calls subroutine MEASR to compute the measured values. Calls subroutine PROCES, which in turn calls the individual transfer and inventory routines.

MESIN. Reads and echo prints all input data.

MTFIX. Calculates components of the variances and covariances as a function of the measurement type.

OUTDEC. Writes output file to DECANAL for the case where both the transfers and inventories are reduced to one component each.

PROCES. Calls the individual process routines (COLUMN, INV1, INV2, TRAN1, TRAN2, TRAN3, and TRAN4) for computing components and measurements.

PRTBAL. Computes materials balances and associated variances and prints a table of important variables calculated by the MEASIM code.

READEM. Reads measurement error and type information from the input data file for each process variable.

REDPV. Reads the input array of process variables.

RNORM. Generates a deviate from a standard normal distribution.

SETMAS. Computes the variance and correlation indicator arrays associated with each transfer and inventory.

STNDEV. Computes the CUSUM sample variance for a given number of runs. Prints a table summarizing the Monte Carlo simulation.

TRAN1. Computes the transfer component and measurement for a transfer consisting of one measured value.

TRAN2. Computes the transfer component and measurements for a transfer consisting of the product of two measured values.

TRAN3. Computes the transfer components and measurements for a transfer consisting of the product of flow rate and concentration.

TRAN4. Computes the transfer components and measurements for the case when the transfer is modeled as the product of concentration and the difference between a full weight and an empty weight.

WRTB. Writes to the output file (DEBUG mode only) the arrays of correlated error variances.

WRTC. Writes to the output file (DEBUG mode only) the arrays of correlation indicators.

WRITEM. Echo prints from the input file the measurement error and type data associated with each process variable.

WRTR. Writes to the output file (DEBUG mode only) the arrays of uncorrelated error variances.

WRTS. Writes to the output file (DEBUG mode only) the arrays of sequential correlation indicators.

WRT3 and WRT4. Writes to the output file (DEBUG mode only) the covariance matrices computed in subroutine MASAGE.

ZERO. Sets to zero the inventories, transfers, and associated variances.

B. Block Diagram

A block diagram showing the interaction of the important subroutines in the code is shown in Fig. 1. Each of the blocks in the diagram represents a subroutine.

After opening the necessary input and output files, the MAIN program calls MESIN to read in and echo print to output the input data. MESIN calls SETMAS, BLANKS, READEM, REDPV, and WRITEM to assist in this function. SETMAS computes the measurement error variances and the correlation indicators for each transfer and

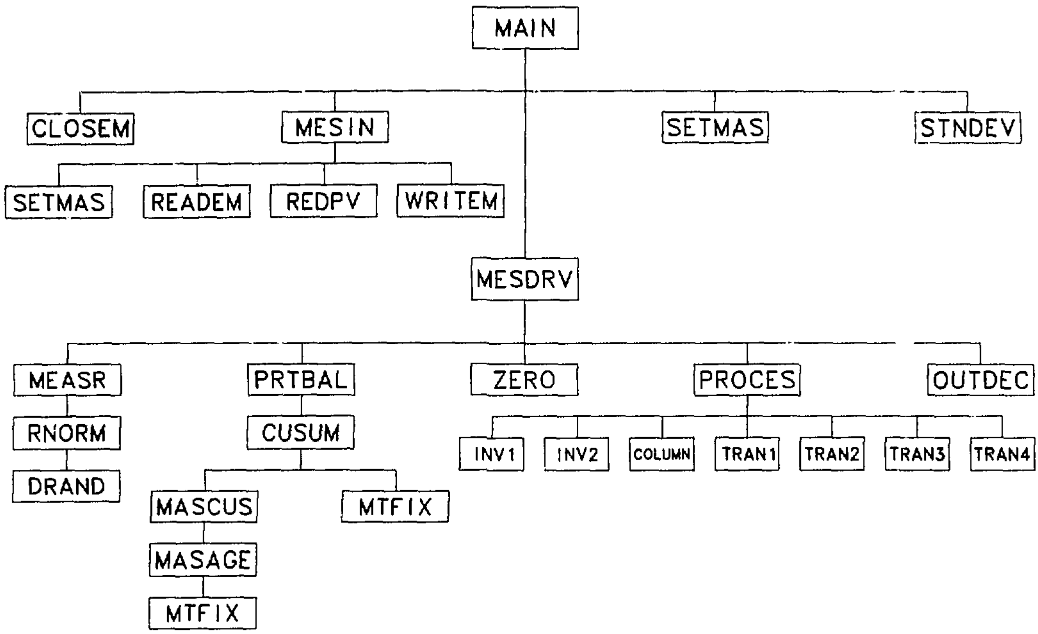

Fig. 1.
MEASIM code block diagram.

inventory measurement in the unit process accounting area. These variances are required in subroutine MASAGE and for input to the DECANAL code.

With the initializations completed, MAIN transfers control to subroutine MESDRV. A call to ZERO from MESDRV sets the inventories and transfers to zero. The individual process variable measurement errors are calculated by calls to subroutine MEASR. MEASR calls RNORM, the standard normal deviate generator, and RNORM calls DRAND to generate uniform random numbers in the interval 0-1. Next MESDRV repeatedly calls subroutine PROCES for the individual inventory and transfer calculations. PROCES calls the appropriate process routine, INV1, INV2, COLUMN, TRAN1, TRAN2, TRAN3, or TRAN4.

At this point in the computation, the measurement code MEASIM has essentially satisfied the primary goal, namely to generate a set of measured values from the input process variables. The

remaining MEASIM calculations are useful for checkout and for reducing the dimensions of the subsequent DECANAL calculations.

The materials balances, CUSUM, and CUSUM variance calculations are important for verifying the correctness of the measurement model. These calculations are initiated with MESDRV calling subroutine PRTBAL.

Materials balances are calculated in PRTBAL, and PRTBAL calls subroutine CUSUM to calculate the CUSUM and CUSUM variance. Subroutines MASCUS and MASAGE are used for the variance and covariance calculations associated with the CUSUM variance. Before returning control to the MAIN program, MESDRV may call subroutine OUTDEC to write an output file to DECANAL. OUTDEC is called only if a reduction of both the inventories and transfers to single inventory and transfer components is desired.

MESDRV transfers control back to MAIN where a check is made for the Monte Carlo option. This option serves as a very useful check on the measurement and variance calculations. If the Monte Carlo option has been selected, MESDRV is called again to repeat the measurement calculations. The measurements will be different in this case because the random numbers have changed. Subroutines PRTBAL and CUSUM are called to calculate the CUSUM, but the variances calculations are not repeated. This procedure is repeated until the desired number of runs is completed. STNDEV is then called to compute the sample CUSUM variance at each materials balance and to output a summary table for the Monte Carlo simulation.

Before terminating the run, MAIN checks for more data. If more data are present, another run is made. This procedure is repeated until all the data have been read. After all the data have been exhausted, MAIN calls subroutine CLOSEM to close the files and terminate the run.

C. Array Sizes and the FORTRAN PARAMETER Statement

Most of the arrays in the code are dimensioned with the aid of a FORTRAN "PARAMETER" statement. This feature makes it easy to increase or decrease array sizes as required by the particular

process being modeled. With those computer systems that do not support the PARAMETER statement, it is necessary to dimension the arrays directly with actual integers.

The parameters in the PARAMETER statement are used primarily for setting the dimensions of the arrays. The only other places in the code where the parameters are used are in some DO-loop indices at the beginning of subroutine SETMAS and in the calling arguments to subroutines WRTB, WRTC, WRTR, WRTS, WRT3, and WRT4 that occur in subroutines MASAGE and SETMAS. Hence, if the PARAMETER statement cannot be used, it is only necessary to replace the PARAMETERs with numerical values in (1) statements that define the array sizes, (2) DO-loop indices at the beginning of subroutine SETMAS, and (3) calling arguments of subroutines WRTB, WRTC, WRTR, WRTS, WRT3, and WRT4 in the subroutines MASAGE and SETMAS. This limited use of the PARAMETERs makes adapting the MEASIM code to computer installations not supporting the PARAMETER statement easy.

The PARAMETER statement in the current version of the code is

```
PARAMETER (NBALMX = 105, NBMXP1 = NBALMX + 1,
   NBMXP2 = NBALMX + 2, NBMXP3 = NBALMX + 3, NTRNMX = 515,
   NTMXP1 = NTRNMX + 1, NTMXP2 = NTRNMX + 2, NPVMX = 12,
   NPVIMX = 10, NPVTMX = 8, NCIMX = 10, NCTMX = 8,
   NMIMX = 2, NMTMX = 2, NSIMX = 2, NSTMX = 2, NCMX = 2,
   MXPMX = 2, NSTRMX = 20, NCOLMX = 2)  .
```

These PARAMETER values are sufficiently large to support all the example problems appearing in this manual. The variables appearing in the PARAMETER statement are summarized in Table I.

D. Units

The set of units in which the MEASIM code computes is determined by the units used for the input process variables. For the

TABLE I

VARIABLES IN THE FORTRAN PARAMETER STATEMENT

Variable	Definition	Current Value
NBALMX	Maximum number of materials balances	105
NBMXP1	NBALMX + 1	106
NBMXP2	NBALMX + 2	107
NBMXP3	NBALMX + 3	108
NTRNMX	Maximum number of transfer values for each transfer process variable	515
NTMXP1	NTRNMX + 1	516
NTMXP2	NTRNMX + 2	517
NPVMX	Maximum number of process variables	12
NPVIMX	Maximum number of inventory process variables	10
NPVTMX	Maximum number of transfer process variables	8
NCIMX	Maximum number of inventory components	10
NCTMX	Maximum number of transfer components	8
NMIMX	Maximum number of inventory measurements per component	2
NMTMX	Maximum number of transfer measurements per component	2
NSIMX	Maximum number of correlated errors per inventory measurement	2
NSTMX	Maximum number of correlated errors per transfer measurement	2
NCMX	Minimum number of NMIMX and NMTMX	2
MXPMX	Maximum number of NSIMX and NSTMX	2
NSTRMX	Maximum number of random-number streams	20
NCOLMX	Maximum number of pulsed columns	2

example problems appearing in this manual, the input process variables have units in kilograms, liters, and hours. Hence, the outputs from MEASIM will have similar units.

E. Input/Output Files

The MEASIM code requires five files for input/output purposes. These files and their associated logical-unit numbers are summarized in Table II. Content and format details of these files will be given in later sections of this manual.

III. FUNDAMENTALS OF MEASUREMENT MODELING

Before discussing the more detailed aspects of the MEASIM code, it will be helpful to present some of the fundamental concepts on which the code is based.

A. Unit Process Accounting Area

The overall process can be considered to consist of one or more distinct, individual processes called unit processes. As an example, consider the hypothetical process shown in Fig. 2. In this case the complete process consists of five unit processes including the surge tank, pulsed column, feed tank, catch tank, and sample tank. A Unit Processing Accounting Area (UPAA) consists of one or more unit processes around which it is desired to

TABLE II

INPUT/OUTPUT FILE SUMMARY

Logical Unit No.	Fortran Symbol	Description
11	NDAT	Input data
12	NPVIN	Input process variable
13	NFSEED	Input random-number seeds
14	NDECIN	Output to DECANAL
15	NPROUT	General descriptive MEASIM output

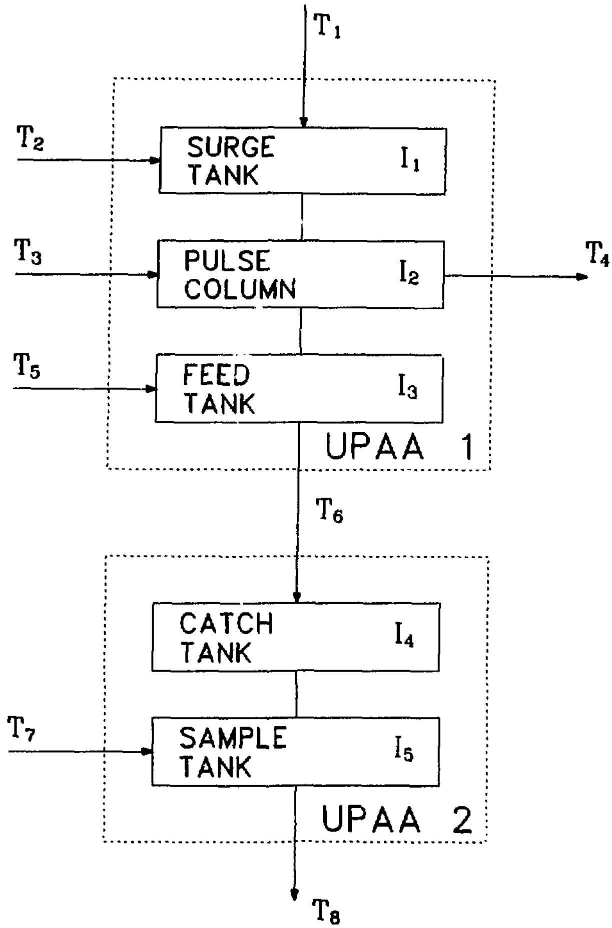

Fig. 2.
Example process.

do materials accounting. The process of Fig. 2 is divided into the two unit process accounting areas, UPAA1 and UPAA2, defined by the dashed lines. UPAA1 contains the surge tank, pulsed column, and feed tank, and UPAA2 contains the catch tank and sample tank.

B. Process Variables

From the standpoint of the MEASIM code, a process variable is any variable needed for the computation of the inventories and transfers for a given UPAA. Examples of process variables are the flow rate of an input stream, volume of a tank, concentration of a tank, etc. The MEASIM code simulates the measured values of process variables. It is not necessary, however, that all process variables have associated measured values.

C. Transfers and Inventories

UPAA1 has six different paths for transferring Special Nuclear Material (SNM) across the boundaries (Fig. 2). Transfers between elements within a UPAA are not important for materials accounting. Therefore, for materials accounting purposes, UPAA1 has a total of six transfers. Each separate block in UPAA1 defines an inventory. Thus, UPAA1 has a total of three inventories. Similarly, UPAA2 has three transfers and two inventories.

The MEASIM code combines the individual transfers and inventories to model a given UPAA. Each transfer and inventory type has an associated identification number. These numbers are used in formulating input data to build the desired UPAA. The inventory and transfer types currently available in the code along with their corresponding identification numbers are summarized in Table III. The identification numbers for transfers, i.e., 5, 6, 7, and 8 are also used to indicate transfer direction. Positive numbers are used for input transfers and negative numbers for output transfers.

All the inventories and transfers required to model a given UPAA are combined to form an inventory-transfer set. The number

TABLE III

INVENTORY AND TRANSFER SUMMARY

No.	Type
1	Inventory - measured directly
2	Inventory - product of volume and concentration
3	Inventory - type "A" pulsed column
4	Inventory - type "B" pulsed column
5	Transfer - measured directly
6	Transfer - product of volume and concentration
7	Transfer - product of flow rate and concentration
8	Transfer - product of concentration and a change in mass

of inventories and transfers in the set and the specific inventory and transfer types are defined with input data.

D. Components and Measurements

Inventories and transfers are constructed from components and measurements. Each inventory and transfer can be formed from the sum of one or more components. A component can be defined as the smallest entry into which an inventory or transfer can be divided and still retain the dimensions of an inventory or transfer. As an example, if a transfer is computed from the product of volume and concentration, then this product is a transfer component. Inventory and transfer components are functions of the measured values of the process variables. In this code, measurements are defined as the measured value of a process variable multiplied by a suitable constant. The product of flow rate and a time interval is an example of a measurement. In this example of a transfer computed as the product of measured values of volume and concentration, both volume and concentration are considered as measurements. In most cases, components are either directly equal to a measurement or to the product of two measurements.

Components and measurements associated with each of the transfer and inventory types currently used in the code are summarized in Table IV, which shows that the pulsed column inventories are computed from the sum of four components. The first three components are measurements computed from the product of a constant and a measured value of concentration, while the fourth component is a collection of terms not dependent upon any measurements. These fourth components for all the pulsed columns in a given UPAA are summed in the same inventory component.

Four transfer components are required to model the transfer computed from the product of flow rate and concentration. In this case each component is defined by the product of two measurements, namely the flow rate and the product of a constant and the concentration. The subscript "1" indicates measured process variables at the previous time, and "2" indicates values at the current time.

TABLE IV

COMPONENTS AND MEASUREMENTS FOR INVENTORIES AND TRANSFERS[a]

Inventory or Transfer No.	Type	Component No.	Measurement 1	Measurement 2
1	Inventory - direct	1	I	
2	Inventory - product of volume and concentration	1	V	C
3 or 4	Inventory - pulse columns	1	K_1C_F	-
		2	K_2C_W	-
		3	K_3C_P	-
		4	Remainder	-
5	Transfer - direct	1	T	
6	Transfer - product of volume & concentration	1	V	C
7	Transfer - product of concentration & flow rate	1	F_1	K_4C_1
		2	F_2	K_4C_2
		3	F_2	$K_4C_1/2$
		4	F_1	$K_4C_2/2$
8	Transfer - product of concentration & a change in mass	1	C	M_i
		2	C	M_f

[a]Notation

I - inventory
V - volume
C - concentration
T - transfer
M_i - initial mass
M_f - final mass
K_i - constant
C_F - feed conc.
C_W - waste conc.
C_P - product conc.
F_1 - previous flowrate
F_2 - current flowrate
C_1 - previous conc.
C_2 - current conc.

E. Materials Balance and CUSUM

In performing materials accounting for a given UPAA, computing a materials balance for that UPAA is usually necessary. A materials balance calculation must be associated with a specific time interval. By definition the Materials Balance (MB) over the time interval t_{i-1} to t_i is

$$MB(t_{i-1},t_i) = I(t_{i-1}) - I(t_i) + T(t_{i-1},t_i) \quad , \tag{1}$$

where

$I(t_i)$ = material inventory in UPAA at time t_i, and
$T(t_{i-1},t_i)$ = net material transfer into the UPAA over the time interval t_{i-1} to t_i.

If there is a net transfer of material out of the UPAA, then the sign of T will be negative.

For example, consider a materials balance calculation for the UPAA1 of Fig. 2 with the inventories represented by I_1, I_2, and I_3 and the transfers by T_1, T_2, T_3, T_4, T_5, and T_6. The total inventory I for the UPAA1 is

$$I = I_1 + I_2 + I_3 \quad ,$$

and the net input transfer T is

$$T = T_1 + T_2 + T_3 - T_4 + T_5 - T_6 \quad .$$

The materials balance can then be calculated by direct substitution into Eq. (1).

The CUmulative SUMmation (CUSUM) of materials balances is defined over a time interval spanning one or more materials balance intervals and is equal to the sum of the materials balances over that interval. In the general case where N materials balance intervals span the interval t_o to t_N, the CUSUM will be

$$\text{CUSUM}\ (t_o, t_N) = \sum_{i=1}^{N} \text{MB}(t_{i-1}, t_i) \quad .$$

Substituting MB from Eq. (1) into the above equation gives

$$\text{CUSUM}(t_o, t_N) = I(t_o) - I(t_N) + \sum_{i=1}^{N} T(t_{i-1}, t_i) \quad . \tag{2}$$

As indicated in Sec. I, the materials balance and CUSUM calculations serve mainly as valuable debugging aids.

F. Measurement Error Models

The primary purpose of the MEASIM code is to model measurement errors and, hence, compute measured values for process variables. Measurement errors are considered to be of two types, correlated and uncorrelated, sometimes called systematic and random. Correlated errors are further classified as either short- or long-term. Uncorrelated errors change each time a measurement is made. Short-term correlated errors remain constant until the measuring instrument is recalibrated, whereas long-term correlated errors remain constant over the entire simulation. For each measurement, the MEASIM code allows for three errors; an uncorrelated error, a short-term correlated error, and a long-term correlated error. All errors are generated as deviates from zero-mean, normal distributions. Subroutine RNORM generates deviates from a standard normal distribution (zero mean and a standard

deviation of one). The resulting random number from RNORM is multiplied by the standard deviation of the measurement error to obtain the actual error.

The measurement model defines the way the actual process values and measurement errors are combined in computing measured values. The following three error models are currently available in the MEASIM code:

$$\text{multiplicative:} \quad m = \mu(1 + \varepsilon + \eta + \theta) \quad ; \tag{3}$$

$$\text{additive:} \quad m = \mu + \varepsilon + \eta + \theta \quad ; \tag{4}$$

$$\text{mixed:} \quad m = \mu(1 + \varepsilon + \eta) + \theta \quad ; \tag{5}$$

where

m = measured value,

μ = true process value,

ε = uncorrelated error,

η = short-term correlated error, and

θ = long-term correlated error.

G. Variances and Covariances

The primary purpose of MEASIM is to compute measured values from true values and measurement errors and write these measured values to a file that can be read by the decision analysis code, DECANAL. However, MEASIM also performs other auxiliary calculations that necessitate the computation of variances and covariances. In the normal operation mode each transfer and inventory measurement, along with the measurement error standard deviations, are output by MEASIM to DECANAL. However, for some processes, very little accuracy is lost by first combining the inventories and transfers into a single inventory and a single transfer along with the equivalent variances and covariances and then writing this information to a file for input to DECANAL. This procedure

can significantly reduce the dimensions of the problem and result in much lower computation times. Another case where MEASIM uses variances and covariances is in the CUSUM variance calculation. The CUSUM variance serves as a very powerful tool for checking measurement calculations. Because DECANAL also computes the CUSUM variance, the variances calculated from the two codes should, for most cases, be the same.

Because of this need for calculating variances and covariances, the main tabular output from MEASIM contains inventory and transfer variances and covariances as well as the CUSUM variance. For a given random variable X with mean μ_x, the variance is defined by

$$\mathrm{Var}(X) = E\{(X - \mu_x)^2\} \quad , \tag{6}$$

where $E\{\}$ is the "expected value" operator. An alternate notation for variance of X is σ_x^2. Similarly, the covariance of two random variables X and Y with means μ_x and μ_y, respectively, is defined by

$$\mathrm{COV}(X,Y) = E\{(X - \mu_x)(Y - \mu_y)\} \quad . \tag{7}$$

It is beyond the scope of this user's manual to derive the variance and covariance calculations performed in MEASIM. Individual transfer and inventory variances and covariances are computed in subroutine MASAGE. Subroutine SETMAS, which serves as a pre-processor for MASAGE, computes individual error variances and the correlation indicators required by MASAGE. Subroutine MASCUS operates on the MASAGE output to compute equivalent inventory and transfer variances and covariances. Subroutine CUSUM also uses MASAGE output to compute the CUSUM variance.

H. Inventory Correlation Approximations

If a measurement is common to two different inventories, then these inventories are correlated because they both contain the same error (see Appendix A). Similarly, if an inventory and transfer contain the same measurement, then the inventory and transfer are correlated. Appendix A shows that these correlations will have a relatively small effect upon the CUSUM variance under steady-state operating conditions. Therefore, the MEASIM code does not include the effects of these correlations in the calculation of the CUSUM variance. In addition, although the code does include the effects of the correlated errors when calculating the inventory variance, Appendix A shows that this inventory variance also has a small effect upon the CUSUM variance. Therefore, under steady-state operating conditions, the inventory correlated errors can be assumed to be zero.

IV. INPUT DATA

The MEASIM code has been developed so that all process-dependent information is handled through input. No coding changes should be required when applying the code to different processes. The only exception might be a change in the PARAMETER statement to adjust the dimensions of the arrays to meet the needs of a particular application.

Three input files are required to run the code: (1) primary input file, (2) process variable file, and (3) random-number seed file. The primary input file defines the UPAA, the measurement errors and measurement model, and other information pertinent to the simulation. The process variable file contains at a minimum those process variables from the MODEL code that are necessary to perform a materials balance calculation for the UPAA, and the random-number seeds file contains the numbers necessary for starting the random-number generator.

A. Primary Input File

Table V gives a complete listing of all possible input data to the code. The input is free-format. A "/" is used to terminate an input data line. Input variables that do not appear on the input file take on their corresponding default values. A "/" in the first column is frequently used to indicate a comment line. The data sequencing and the comment lines must be adhered to strictly. An explanation of all input variables and their default values is given in Table VI.

The specific process is defined by the NPV, NTRIN, ITIN, IPVNO, and ISPNTI inputs. NPV is equal to the number of different process variables read in from the process variable file. In general, NPV must be greater than or equal to the number of process variables required for the UPAA of interest. NTRIN specifies the number of transfers and inventories required to define the UPAA. Specific inventory and transfer types are defined by the ITIN vector as described in Table VI. The ordering of inventories and transfers in the ITIN vector is optional, but this ordering is used to assign a number to each inventory or transfer. Hence, if a transfer is specified in the Mth element of ITIN, this transfer is assigned an inventory-transfer number of M.

The process variables corresponding to each inventory or transfer are selected with the IPVNO array. Each process variable has a number determined by the order that it is read in from the process variable file. A process variable is selected for a given inventory or transfer by inserting this number in the IPVNO array. Each line of input data for the IPVNO array contains the process variable numbers for one inventory or transfer. A maximum of five numbers can be used for any one line. The ordering of the IPVNO input data lines must correspond to the sequence of inventories and transfers established by the ITIN vector. For example, if the third entry in ITIN specifies a column inventory, then the third line of input data for IPVNO must specify the process variables for the column inventory calculation. Each line of the IPVNO input contains the process variable numbers

TABLE V

INPUT DATA

```
/    This is a summary of the input data to the MEASIM code.
/    The 6 lines shown here beginning with a "/" are used for
/    comments.
/
/
/
SIMULATION IDENTIFIER
/
IZE
IRNSCH
NRUN
NBAL
NTRPBL
DT
MASPRT
ITIPRP
IMESPR
IPVPRT
ICLAPS
/
/    Input data for defining unit process accounting area
/
NPV
NTRIN
/
/
ITIN(1),ITIN(2),...,ITIN(NTRIN)
/
/
IPVNO(1,1),IPVNO(2,1),...,IPVNO(5,1)
IPVNO(1,2),IPVNO(2,2),...,IPVNO(5,2)
.    .    .    .
.    .    .    .
IPVNO(1,NTRIN),IPVNO(2,NTRIN),...,IPVNO(5,NTRIN)
/
ISPNTI
/
/
/
1:   Identifier for first process variable
SIGMAE(1),SIGMAN(1,1),SIGMAN(1,2),MESTYP(1),INTCAL(1,1),INTCAL(1,2)
2:   Identifier for second process variable
SIGMAE(2),SIGMAN(2,1),SIGMAN(2,2),MESTYP(2),INTCAL(2,1),INTCAL(2,2)
.    .    .    .
.    .    .    .
NPV:  Identifier for NPVth process variable
SIGMAE(NPV),SIGMAN(NPV,1),SIGMAN(NPV,2),MESTYP(NPV),INTCAL(NPV,1),INT
CAL(NPV,2)
/
/    Column Data - one line per column
/    This data including these four comment lines are not present
/    when no columns are modeled.
(HCC(I,1),I=1,5), (VCC(I,1),I=1,2), (CCC(I,1),I=1,3)
(HCC(I,2),I=1,5), (VCC(I,2),I=1,2), (CCC(I,2),I=1,3)
.    .    .    .
.    .    .    .
(HCC(I,NCOLUM),I=1,5), (VCC(I,NCOLUM),I=1,2), (CCC(I,NCOLUM),I=1,3)
ISPNTI
ISPNTI   This last set of "ISPNTI" inputs indicates to the code
....     that the run is to be repeated for specific inventory
....     transfer numbers.
ISPNTI
```

TABLE VI

EXPLANATION OF INPUT VARIABLES AND DEFAULT VALUES

Variable	Description	Default Value
IZE	All measurement errors are set to zero when IZE = 1	0
IRNSCH	Random-number seeds change between simulations when IRNSCH = 1	0
NRUN	Number of runs. If NRUN>1 a Monte Carlo simulation is assumed	1
NBAL	Number of materials balances to be computed	1
NTRPBL	Number of transfers per materials balance	1
DT	Time interval between transfers (meaningful only if transfer is computed as product of flowrate and concentration)	1
MASPRT	Provides detailed printout of MASAGE calculations when MASPRT = 1	0
ITIPRP	Provides transfer and inventory number printout when ITIPRP = 1	0
IMESPR	Echo prints measurement error data when IMESPR = 1	0
IPVPRT	Provides printout of the input process variable array when IPVPRT = 1	0
ICLAPS	Flag for reducing dimensions of the output file written to DECANAL code 0 - no reduction 1 - all inventories reduced to one inventory with single variance and covariance 2 - all inventories and transfers reduced to one inventory and one transfer, each with one equivalent variance and covariance	0
NPV	Number of process variables	1

TABLE VI (cont)

Variable	Description	Default Value
NTRIN	Number of transfers and inventories in UPAA	1
ITIN(I)	Selector for the Ith inventory or transfer 1 - Inventory as one measured value 2 - Inventory as product of two measured values 3 - Inventory as a type "A" pulsed column 4 - Inventory as a type "B" pulsed column 5 - Input transfer as one measured value 6 - Input transfer as the product of two measured values 7 - Input transfer as the product of flow rate and concentration 8 - Input transfer as the product of concentration and the difference between an initial and final weight -5 - Output transfer as one measured value -6 - Output transfer as the product of two measured values -7 - Output transfer as the product of flow rate and concentration -8 - Output transfer as the product of concentration and the difference between an initial and final weight	
IPVNO(I,J)	Process variable number for the Ith process variable in the Jth inventory or transfer	0
ISPNTI	Specific transfer or inventory set to be calculated. When ISPNTI = 0, all the transfers and inventories for the UPAA are calculated	0
SIGMAE(I)	Standard deviation of the uncorrelated error for the Ith process variable	0
SIGMAN(I,1)	Standard deviation of the short-term correlated error for the Ith process variable	0
SIGMAN(I,2)	Standard deviation of the long-term correlated error for the Ith process variable	0

TABLE VI (cont)

Variable	Description	Default Value
MESTYP(I)	Measurement error model type selector for the Ith process variable 1 - multiplicative model 2 - additive model 3 - multiplicative for random and short-term correlated errors: Additive for long-term correlated errors	1
INTCAL(I,1)	Number of transfers between recalibrations for the short-term correlated errors	10 000
INTCAL(I,2)	Number of transfers between recalibrations for the long-term correlated errors	10 000
HCC(I,J)	Ith holdup constant for the Jth pulse column	None
VCC(I,J)	Ith volume constant for the Jth pulse column	None
CCC(I,J)	Ith concentration constant for the Jth pulsed column	None

corresponding to a transfer or inventory. Only in the case of pulse-column inventories (ITIN = 3 or 4) or for a transfer as the product of a concentration and a mass change, (ITIN = 8), is the order of the numbers important. For pulsed columns, the process variable specified by a line of IPVNO data input must define the process variables in the following order:

(1) feed concentration,

(2) waste concentration,

(3) product concentration,

(4) top organic volume, and

(5) bottom aqueous volume.

For transfers calculated from the product of a concentration and the change in weight the IPVNO input must define the process variables in the following order:

(1) concentration,
(2) full weight, and
(3) empty weight.

For all other transfers and inventories, the sequence for defining the process variables with the IPVNO input is unimportant. As an example, if an inventory is calculated from the product of volume and concentration, the order for specifying the volume and concentrations process variables in the corresponding IPVNO input data line is optional.

The ISPNTI input variable defines the specific inventory or transfer to be computed. Each inventory or transfer has a number assigned to it by the order in which it is defined by the ITIN input vector. For example, if performing isolated measurement calculations on a specific inventory is desired, and this inventory appears as the Mth element in the ITIN input, then ISPNTI would be set to M. If including all the inventories and transfers in the UPAA is desired, then ISPNTI should be set to 0.

Each NPV process variable requires a set of measurement error data that include the standard deviation for uncorrelated and correlated errors, measurement type, and recalibration intervals. The measurement data for each process variable must appear in the same order that the process variables are numbered. This numbering is established from the process variable sequencing on the input process variable file.

B. Input Process Variable Array

Except for the first (title) line, the process variable file is an unformatted or binary file. The contents of this file, as written by the process model code, is given in Table VII. TITLE can have a maximum of 80 characters. In normal operation, TITLE will be read in on the input data file for the process model code and then written to the process variable file for input to the MEASIM code. The remaining entries on the process variable file are unformatted and are as follows:

N - number of different process variables on the file,

IPVTRN(I) - transfer indicator flag for the Ith process variable,

0 - inventory process variable,

1 - transfer process variable,

NPVC(I) - number of entries for the Ith process variable,

PV(I,J) - Ith entry for Jth process variable.

The number of different process variables transmitted to MEASIM can be larger than the number of process variables required for the UPAA materials balance. As an example, the process model code could generate a process variable file containing all the process variables of interest for the complete process. Only a portion of these process variables may be used by the MEASIM code for any single UPAA. However, the number of process variables defined by the input data variable NPV must be equal to the number of different process variables on the process variable file. If an inconsistency exists among these process variable numbers, an error message is printed and the run is terminated.

The order of occurrence of the process variables on the input process variable file defines the numbering of the process variables (Sec. III). The input data file must be structured to be consistent with this ordering of the process variables.

TABLE VII

CONTENTS OF PROCESS VARIABLE FILE

```
TITLE
N
IPVTRN(1),IPVTRN(2),...IPVTRN(N)
NPVC(1),NPVC(2),....,NPVC(N)
PV(1,1),PV(2,1),...PV(NPVC(1),1)
PV(1,2),PV(2,2),...PV(NPVC(2),2)
.          .   .   .
.          .   .   .
PV(1,N),PV(2,N),...,PV(NPVC(N),N)
```

Two types of process variables are those associated with transfer calculations and those with inventory calculations. Inventory process variables are required only at materials balance times, but transfer process variables are required any time a transfer is made. Usually, the number of transfer process variables will be at least equal to or greater than the number of inventory process variables. In some isolated instances, a process variable is used for both transfer and inventory calculations. In this case, the process variable should be considered as a transfer on the process variable file.

The minimum number of entries on the process variable file for each transfer and inventory process variable is a function of the number of required materials balances and the number of transfers per materials balance interval. With the number of materials balances given by NBAL and the number of transfers per materials balance interval by NTRPBL, the minimum number of entries for each inventory and transfer process variable are given by

$$NMININ = NBAL + 2 \quad , \text{ and} \tag{8}$$

$$NMINTR = (NBAL + 1)NTRPBL + 3 \tag{9}$$

where

NMININ = minimum number of entries for each inventory process variable, and

NMINTR = minimum number of entries for each transfer process variable.

The number of entries can be larger than, but not smaller than, the above values. If the numbers of transfer or inventory entries are less than the required minimum, the code prints an error message and the run is terminated.

C. Input Random-Number Seeds

The random-number seeds that serve as starters for the uniform random-number generator, DRAND, are input to the code on a separate file. Each nonzero measurement error, uncorrelated or correlated, requires a separate random-number seed. The actual number of random-number seeds required for a given simulation is NNSTRM, with NNSTRM computed by the code in subroutine SETMAS. As a consequence, the input random-number seed file must contain at least NNSTRM numbers. The input random-number seeds are ten-digit integers with a maximum value of 2 147 483 648 (2^{31}). The NNSTRM numbers must appear on the input random-number seed file with one number per line.

If the input integer, IRNSCH, in the primary input file is set to 1, then the random-number seed file will be changed at the end of each simulation. The random-number seed file will remain unchanged if IRNSCH = 0.

The maximum number of random-number seeds that the program can handle is determined by NSTRMX in the FORTRAN PARAMETER statement. In the current version of the code NSTRMX is set at 10.

V. OUTPUTS

The MEASIM code generates two output files, the primary output and the output to the decision analysis code, DECANAL. The primary output contains an echo check of the input and the results of the simulation, whereas the output to DECANAL contains all the required measured process values, measurement error standard deviations (σ's), and correlation indicators.

An example of the primary output file is given in Fig. 3. The first three lines of the output file indicate the file names for the input data, the input process variables, and the output to the DECANAL code. Two descriptor lines, one from the process

```
INPUT DATA FILE - MESDT1
PV ARRAY FILE - PVARA1
OUTPUT DECANAL FILE - DECINN

MON, OCT 19 1981      14:49:44

TITLE FROM PROCESS MODEL CODE

EXAMPLE 1 - SAMPLE PROBLEM 1 FROM PROCESS MODEL USER'S MANUAL (LA-8761-M)

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = 0
    NUMBER OF RUNS (NRUN) = 100
    NUMBER OF BALANCES (NBAL) = 50
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) =  1
    TIME INTERVAL (DT) =   0.000
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = 0
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = 0
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = 0
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = 0
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. 0) = 0


****************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) =  8
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =  5
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      6  2  2 -5 -5

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY            PROCESS VARIABLE NUMBER
          NUMBER       (1)     (2)     (3)     (4)     (5)
            1           1       2       0       0       0
            2           3       4       0       0       0
            3           5       6       0       0       0
            4           7       0       0       0       0
            5           8       0       0       0       0

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  0

****************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY =  8
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 1 0 0 0 0 1 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
      55  55  53  53  53  53  55  55

READING OF PROCESS VARIABLE ARRAY COMPLETE
****************************************************

MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE

    1:  INPUT VOLUME
        INITIAL VALUE =   100.000000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.010000    0.000000       1        10000        10000

    2:  INPUT CONCENTRATION
        INITIAL VALUE =      0.050000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.010000    0.000000       1        10000        10000

    3:  TANK1 VOLUME
        INITIAL VALUE =   100.000000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.000000    0.000000       1        10000        10000

    4:  TANK1 CONCENTRATION
        INITIAL VALUE =      0.050000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.000000    0.000000       1        10000        10000

    5:  TANK2 VOLUME
        INITIAL VALUE =   100.000000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.000000    0.000000       1        10000        10000

    6:  TANK2 CONCENTRATION
        INITIAL VALUE =      0.050000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.000000    0.000000       1        10000        10000

    7:  WASTE PU
        INITIAL VALUE =      0.010000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.010000    0.000000       1        10000        10000

    8:  PRODUCT PU
        INITIAL VALUE =      4.990000
                  SIGMAE    SIGMAN(1)   SIGMAN(2)    MESTYP    INTCAL(1)     INTCAL(2)
                0.010000   0.010000    0.000000       1        10000        10000
```

Fig. 3.
Example of primary output file.

```
*******************************************************

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

    NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) =  8
    NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) =  4
    NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  4
    NUMBER OF INVENTORY COMPONENTS (NCI) =  2
    NUMBER OF TRANSFER COMPONENTS (NCT) =  3
    NUMBER OF INVENTORY MEASUREMENTS (NMI) =  2
    NUMBER OF TRANSFER MEASUREMENTS (NMT) =  2
    NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
    NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  1
    NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) = 12
    NUMBER OF PULSE COLUMNS (NCOLUM) =  0

*******************************************************

INITIAL RANDOM NUMBER SEEDS
 1042839332   273785636  1973287924  1291185537  2063984014   164960264   491400187  1352537143   271703562  1141280589
  763526641   225695806

*******************************************************

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

*******************************************************
```

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.0000E 01	1.0000E-02	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-9.5367E-07	3.4980E-02
2	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.9073E-06	6.4940E-02
3	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.8610E-06	1.0988E-01
4	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.8147E-06	1.6980E-01
5	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.7684E-06	2.4470E-01
6	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-5.7220E-06	3.3458E-01
7	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-6.6757E-06	4.3944E-01
8	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-7.6294E-06	5.5928E-01
9	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-8.5831E-06	6.9410E-01
10	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-9.5367E-06	8.4390E-01
11	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.0490E-05	1.0087E 00
12	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.1444E-05	1.1884E 00
13	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.2398E-05	1.3832E 00
14	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.3351E-05	1.5929E 00
15	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.4305E-05	1.8176E 00
16	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.5259E-05	2.0573E 00
17	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.6212E-05	2.3119E 00
18	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.7166E-05	2.5816E 00
19	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.8120E-05	2.8662E 00
20	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.9073E-05	3.1658E 00
21	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.0027E-05	3.4804E 00
22	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.0981E-05	3.8099E 00
23	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.1935E-05	4.1545E 00
24	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.2888E-05	4.5140E 00
25	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.3842E-05	4.8885E 00
26	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.4796E-05	5.2780E 00
27	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.5749E-05	5.6824E 00
28	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.6703E-05	6.1019E 00
29	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.7657E-05	6.5363E 00
30	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.8610E-05	6.9857E 00
31	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.9564E-05	7.4501E 00
32	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.0518E-05	7.9294E 00
33	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.1471E-05	8.4238E 00
34	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.2425E-05	8.9331E 00
35	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.3379E-05	9.4574E 00
36	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.4332E-05	9.9966E 00
37	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.5286E-05	1.0551E 01
38	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.6240E-05	1.1120E 01
39	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.7193E-05	1.1704E 01
40	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.8147E-05	1.2304E 01
41	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.9101E-05	1.2918E 01
42	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.0054E-05	1.3547E 01
43	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.1008E-05	1.4191E 01
44	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.1962E-05	1.4850E 01
45	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.2915E-05	1.5524E 01
46	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.3869E-05	1.6213E 01
47	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.4823E-05	1.6917E 01
48	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.5776E-05	1.7636E 01
49	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.6730E-05	1.8370E 01
50	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.7684E-05	1.9119E 01

Fig. 3. (cont)

```
*********************************************************

 RESULTS OF MONTE CARLO SIMULATION WITH   100 SAMPLES

*********************************************************

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
    UPPER LIMIT =    1.29223
    LOWER LIMIT =  0.736473
```

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	4.890213D-03	3.498001E-02	4.178171E-02	1.19445	OK
2	-1.360767D-02	6.494007E-02	6.673788E-02	1.02768	OK
3	1.511354D-02	0.109880	9.982021E-02	0.908446	OK
4	4.801090D-02	0.169800	0.166844	0.982588	OK
5	2.539335D-02	0.244700	0.264282	1.08002	OK
6	3.461915D-02	0.334580	0.360880	1.07861	OK
7	2.139579D-02	0.439441	0.486010	1.09232	OK
8	3.131196D-02	0.559281	0.627009	1.12110	OK
9	2.952456D-02	0.694101	0.756659	1.09013	OK
10	1.571570D-02	0.843901	0.925680	1.09691	OK
11	4.273133D-02	1.00868	1.13954	1.12973	OK
12	3.167394D-02	1.18844	1.33260	1.12130	OK
13	1.693831D-02	1.38318	1.53002	1.10616	OK
14	4.041709D-02	1.59290	1.79038	1.12398	OK
15	1.979237D-02	1.81760	2.04000	1.12236	OK
16	1.976359D-02	2.05728	2.33121	1.13315	OK
17	2.360842D-02	2.31194	2.65347	1.14773	OK
18	3.092502D-02	2.58158	2.91311	1.12842	OK
19	3.978035D-02	2.86620	3.28614	1.14651	OK
20	2.544737D-03	3.16580	3.52372	1.11306	OK
21	2.926647D-02	3.48038	3.84259	1.10407	OK
22	5.912140D-02	3.80993	4.17124	1.09483	OK
23	4.381735D-02	4.15447	4.48605	1.07981	OK
24	7.532030D-02	4.51399	4.86077	1.07682	OK
25	9.162207D-02	4.88849	5.20704	1.06516	OK
26	7.438610D-02	5.27797	5.62947	1.06660	OK
27	8.601636D-02	5.68243	6.18899	1.08915	OK
28	9.582625D-02	6.10186	6.51722	1.06807	OK
29	9.471298D-02	6.53628	7.07026	1.08170	OK
30	9.509821D-02	6.98568	7.51287	1.07547	OK
31	1.019806D-01	7.45006	7.99398	1.07301	OK
32	1.019388D-01	7.92941	8.29881	1.04659	OK
33	9.548896D-02	8.42375	8.94736	1.06216	OK
34	8.829592D-02	8.93307	9.54147	1.06811	OK
35	1.063070D-01	9.45736	10.1711	1.07546	OK
36	8.405256D-02	9.99664	10.7608	1.07644	OK
37	7.852274D-02	10.5509	11.3129	1.07222	OK
38	9.006483D-02	11.1201	11.9847	1.07775	OK
39	8.929556D-02	11.7043	12.6697	1.00248	OK
40	1.021226D-01	12.3035	13.1582	1.06946	OK
41	9.188520D-02	12.9177	13.9277	1.07818	OK
42	1.144997D-01	13.5469	14.7081	1.08572	OK
43	1.106139D-01	14.1910	15.3298	1.08025	OK
44	1.046790D-01	14.8501	16.0367	1.07990	OK
45	1.130534D-01	15.5242	16.7761	1.08064	OK
46	1.150222D-01	16.2133	17.6229	1.08694	OK
47	1.114169D-01	16.9173	18.3664	1.08566	OK
48	1.412201D-01	17.6364	18.8992	1.07160	OK
49	1.450512D-01	18.3704	19.9447	1.0857[illegible]	OK
50	1.174638D-01	19.1194	20.6565	1.08040	OK

Fig. 3. (cont)

model code and one from the input data follow the date and time printout. The remainder of the output on the first page of Fig. 3 consists mainly of the echo check of the input.

Part of the information from the process variable file is printed after the statement "BEGIN READING IN PROCESS VARIABLE ARRAY." The IPVTRN vector indicates whether a process variable is associated with an inventory or a transfer. A "1" indicates transfer and a "0" indicates inventory. The number of values read in for each process variable also is printed in this region of the output file.

A complete echo check of the measurement error information follows the process variable information. Also included here is the initial value for each process variable as read from the process variable file. The six numbers following the initial value are

SIGMAE - standard deviation for uncorrelated error,

SIGMAN(1) - standard deviation for short-term correlated error

SIGMAN(2) - standard deviation for long-term correlated error,

MESTYP - measurement error model type,
  1 - multiplicative,
  2 - additive,
  3 - mixed,

INTCAL(1) - number of transfers between recalibration of short-term correlated error, and

INTCAL(2) - number of transfers between recalibration of long-term correlated error.

Subroutine SETMAS uses input information defining the UPAA to compute a set of integers that define the dimensions of the problem. These integers appear on the second page of Fig. 3. Frequent use is made of these integers in the subsequent calculations, so the user can verify their correctness.

The initial random-number seeds appear next. These ten-digit integers are read in from a separate file as discussed in Sec. IV.C.

The main output summary table follows at the bottom of p. 2, Fig. 3. Most of the important information concerning the process and the measurements can be found in this table. Each line corresponds to a materials balance. The column headers are defined as follows.

I	- materials balance number
XI	- inventory
S2I	- inventory variance
CVI	- covariance between adjacent inventories
T	- net transfer
S2T	- transfer variance
CVT	- covariance between adjacent transfers
MAT. BAL.	- materials balance
S2XMB	- materials balance variance
CUSUM	- cumulative summation of materials balance (CUSUM)
S2CUSUM	- CUSUM variance.

Variances rather than standard deviations are printed in the table to make quick handchecks much simpler because variances are additive whereas standard deviations are not.

The table on p. 3 of the output file is present only when the code is run in the Monte Carlo mode. This occurs when the number of runs (NRUN) is set greater than 1 on the input file. In this case, the sample CUSUM variance σ^2 is computed at each materials balance and compared with the propagated σ^2 that is calculated directly from the variances and covariances of the inventories and transfers. The CUSUM propagated and sample variances appear in columns 3 and 4, respectively, of the table. Column 5 contains the ratio of the sample to propagated variance. A 95% confidence interval (see Appendix B) is defined for this ratio at the top of this page. If this ratio of sample to propagated variances lies within the confidence interval, then an "OK" appears in the sixth column. If the ratio lies outside the confidence interval, then "***** RATIO OUTSIDE THE INTERVAL" appears in column 6. As discussed in Appendix B, the correlation of the sample CUSUM variances at each materials balance time makes it impossible to conclude that about 95% of the ratios will lie within the 95% confidence interval for a given Monte Carlo simulation. The correct conclusion is that for a sufficiently large number of different Monte Carlo simulations, the ratio at a given balance time will lie within the confidence interval for about

95% of simulations. On any given Monte Carlo simulation, because of the correlations between the ratios, it is possible that the ratios at every balance time lie outside the confidence interval. In these cases, the Monte Carlo simulations should be repeated before drawing any conclusions concerning the correctness of the measurement model and the propagated variances.

The output is concluded with the final random-number seeds and the computer CPU time in seconds.

VI. DEBUGGING AIDS

A number of useful debugging aids have been built into the code to assist the user in case of difficulty. One of the most common sources of error is in the input data. The user either has too much or not enough data, or the data are not in the proper sequence. To assist the user in isolating this problem, a complete echo print of the input data appears on the output file. The input data are printed immediately after being read in, making it easier to locate the input error.

The input flags MASPRT, ITIPRT, and IPVPRT can be set to obtain more detailed output. If MASPRT = 1, then all the variances and covariances computed in subroutine MASAGE and all the variances and correlation indicators computed in subroutine SETMAS are printed to the output file. When ITIPRT is set equal to 1, each inventory or transfer number and the associated process variable number are printed to the output file before the actual calculations for that inventory or transfer are made. Setting IPVPRT to 1 provides a printout of input process variables.

The code also allows the user to perform the measurement model calculations on one isolated inventory or transfer. Hand checking the measurement code calculations as they appear in the output table becomes relatively easy when considering only one isolated inventory or transfer. This inventory/transfer isolation is obtained by setting the input ISPNTI to the desired inventory/

transfer number. In the nondebug mode, ISPNTI = 0 for the combined computation of all the inventories and transfers. Additional ISPNTI entries can be appended at the end of the input data to generate another run with the same input data but a different inventory transfer. In this manner all the individual inventory transfers as well as the combined set can be computed with one execution of the code.

Exceeding the dimensions of an array is one of the most difficult diagnostic problems for a programmer. If an over-extended array destroys part of the code, and the program becomes "lost" during execution, the user can spend hours finding the problem and correcting it. Most of the important array sizes in the code are established by the FORTRAN PARAMETER statement. In most cases, the code calculates the required array sizes based on the problem specified by the input data. If, for any array, the required array size is larger than the actual array size, an error message is printed, and the run is terminated.

The materials balance in column 8 of the primary output file also serves as a useful check on the correctness of the calculations. In most cases, when all the inventories and transfers are being considered, the materials balance should be zero or very close to zero. A nonzero materials balance is usually caused by one of the following:

(a) process variables in error or read incorrectly from input file,

(b) inventory-transfer selector (ITIN) in error on input, and

(c) process variable number (IPVNO) on the input file assigns incorrect process variable to an inventory or transfer.

The Monte Carlo simulation option, which is one of the most useful debugging tools available in the code, is activated by setting the NRUN input integer to the desired number of runs. As indicated in Sec. V, the Monte Carlo simulation generates a sample CUSUM variance (σ^2) for each materials balance and then computes the ratio of the sample σ^2 to the propagated σ^2. If this ratio lies far outside the 95% confidence interval as

determined from the chi-squared distribution (see Appendix B), then there is most likely an error in either the sample or propagated variances. The ratio of the variances for each materials balance along with the limits for the 95% confidence interval appear on the output file when the Monte Carlo option is selected (Sec. V).

If an error is detected when using the Monte Carlo simulation option for the complete UPAA, then the error can be isolated with a Monte Carlo run for each individual inventory and transfer. This can be accomplished as discussed earlier in this section by setting the ISPNTI input to the appropriate inventory-transfer number while at the same time maintaining NRUN at the desired number of runs.

Subroutine SETMAS calculates some important integers that define the dimensions of the simulation. Section V indicates that these integers appear on the output file just before the initial random-number seeds. These integers are computed from the user-supplied input data describing the process. They should be consistent with the UPAA being modeled.

VII. EXAMPLES

The five examples in this section demonstrate most capabilities of the MEASIM code. Each example includes a discussion of the process, the input data, and the corresponding output. All outputs are for the zero-error case wherein the variances and covariances are calculated from input measurement errors, but the measured process values are set equal to the input process variables. Because the process variables are free of process variations, a potential user can easily check his or her version of the code regardless of the computer on which it is implemented. Introducing errors would make duplicating the results on another computing machine almost impossible because of the machine dependency of the random-number generators. For all the examples,

volumes will be in liters (L), concentrations in kilograms per liter (kg/L), flow rates in liters per hour (L/h), and masses in kilograms (kg).

A. Example 1

The process block diagram, which is the same process used for the first example of Ref. 1, is shown in Fig. 4. All the input and output transfers to the two tanks are batched with all transfers taking place simultaneously at 0.5-h intervals. The UPAA is drawn around the entire process.

The input transfer and both tank inventories are computed from the product of volume and concentration. The waste and product outputs are assumed to be measured directly in kilograms of plutonium. The resulting process variables for this example are summarized in Table VIII. Materials balance calculations are performed at 0.5-h intervals in coincidence with the batch transfers.

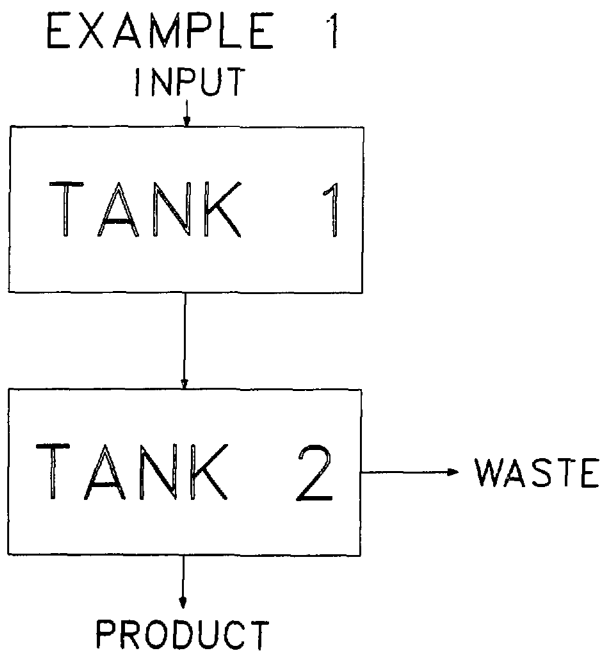

Fig. 4.
Process block diagram--Example 1.

TABLE VIII

PROCESS VARIABLES FOR EXAMPLE 1

No.	Variable	Nominal Value
1	Input volume (L)	100
2	Input concentration (kg/L)	0.05
3	Tank 1 volume (L)	100
4	Tank 1 concentration (kg/L)	0.05
5	Tank 2 volume (L)	100
6	Tank 2 concentration(kg/L)	0.05
7	Waste output (kg)	0.01
8	Product output (kg)	4.99

The input data file for this example is given in Fig. 5 where three transfer components and two inventory components yield a total of five transfer and inventory components. This number appears in line 24 of Fig. 5 for the input variable NTRIN. The five transfer and inventory types are defined in line 27 for the input vector ITIN. Each input defines a transfer or inventory type as defined for the ITIN vector in Table VI. A "6" as the first element of the ITIN vector indicates an input transfer as the product of two measured values, in this case volume and concentration. Entries 2 and 3 for ITIN are both "2," indicating inventories as the product of two measured values. The -5s appearing as the last two entries for ITIN indicate output transfers calculated directly from one measurement. The particular sequence of transfers and inventories in ITIN is by the user's choice and has no effect upon the results of this simulation. However, each transfer and inventory is assigned an inventory-transfer number by virtue of its position in the ITIN array. This number must be strictly adhered to for the subsequent IPVNO and ISPNTI inputs.

```
/ ******* MEASIM INPUT DATA  -  EXAMPLE 1  -  SEPTEMBER 1981
/
/ ******* PARAMETERS (NBALMX=1Ø5, NTRNMX=515, NPVMX=12, NPVIMX=1Ø, NPVTMX=8,
/                    NCIMX=1Ø, NCTMX=8, NMIMX=2, NMTMX=2, NSIMX=2, NSTMX=2,
/                    NCMX=2, MXPMX=2, NSTRMX=2Ø, NCOLMX=2)
/
EXAMPLE 1 - SAMPLE PROBLEM 1 FROM PROCESS MODEL USER'S MANUAL (LA-8761-M)
/
1/            IZE    - ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE)
Ø/            IRNSCH - RANDOM NUMBER SEEDS CHANGE FROM RUN TO RUN WHEN NONZERO
1/            NRUN   - NUMBER OF RUNS
5Ø/           NBAL   - NUMBER OF MATERIALS BALANCES
1/            NTRPBL - NUMBER OF TRANSFERS PER BALANCE
Ø/            DT     - TIME CONSTANT (NOT USED IN THIS EXAMPLE)
Ø/            MASPRT - PRINTS MASSAGE DEBUGG OUTPUT WHEN NONZERO
Ø/            ITIPRP - PRINTS TRANSFER-INVENTORY SET NO. AND PROC. VAR. NO. IF=Ø
Ø/            IMESPR - PRINTS INPUT MEASUREMENT ERRORS WHEN EQUAL TO 1
Ø/            IPVPRT - PRINTS PROCESS VARIABLE FILE WHEN NON ZERO
Ø/            ICLAPS - REDUCES DIMENSIONS OF DECANAL INPUT FILE WHEN NONZERO
/
/ ********** INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA
/
8/            NPV    - NUMBER OF PROCESS VARIABLES
5/            NTRIN  - NUMBER OF TRANSFER AND INVENTORY SETS IN THE PROCESS
/
/   ITIN - INVENTORY-TRANSFER NUMBERS
6 2 2 -5 -5/
/
/   IPVNO - PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER-INVENTORY
1 2/         INPUT TRANSFER
3 4/         TANK1 INVENTORY
5 6/         TANK2 INVENTORY
7/           WASTE OUTPUT
8/           PRODUCT OUTPUT
/
Ø/            ISPNTI - SPECIFIC INVENTORY-TRANSFER SET NO. (Ø GIVES ALL SETS)
/
/ *********** MEASUREMENT ERRORS ASSOCIATED WITH EACH PROCESS VARIABLE
/
1:  INPUT VOLUME
    .Ø1 .Ø1/
2:  INPUT CONCENTRATION
    .Ø1 .Ø1/
3:  TANK1 VOLUME
    .Ø1/
4:  TANK1 CONCENTRATION
    .Ø1/
5:  TANK2 VOLUME
    .Ø1/
6:  TANK2 CONCENTRATION
    .Ø1/
7:  WASTE PU
    .Ø1 .Ø1/
8:  PRODUCT PU
    .Ø1 .Ø1/
```

Fig. 5.
Input data--Example 1.

The IPVNO array defines the process variables associated with each inventory or transfer. Each process variable has a number associated with it as defined in Table VIII. These numbers are used to identify the process variables in the IPVNO input. Each line of IPVNO input corresponds to a transfer or inventory as defined by the ITIN input. The first line of IPVNO data must correspond to the first element of ITIN, etc. In this example, the "6" in the first element of ITIN specifies the input transfer as the product of two measured values. Thus, the first line of IPVNO data contains a "1" and a "2," the process variable numbers

corresponding to this transfer. Similar reasoning applies to the remaining four lines of IPVNO data. The "0" input for ISPNTI at line 36 indicates to the code that all the inventories and transfers defined by ITIN are to be included in the materials balance, CUSUM, and variance calculations.

The input data file is concluded with the measurement errors for each of the process variables. These measurement errors appear in the order defined by the process variable numbering. In this example, all the measurement error standard deviations are set at 0.01. As indicated in Sec. IV, each line of process variable measurement input should contain six entries. In this example the "/" is used to terminate the input measurement error data with a maximum of 1 or 2 entries per line. The omitted entries take on their respective default values as defined in Table VI. As an example, line 41 of the input data assigns a 0.01 standard deviation to the uncorrelated and short-term correlated errors, with the remaining entries for that line taking on default values. This implies a zero for the long-term correlated error, a multiplicative error model, and recalibration intervals for the correlated errors of 10 000 transfers, that is, no recalibration.

The input process variable file must contain eight sets of numbers corresponding to the eight process variables. The number of required transfer and inventory process variables for each set can be computed from Eqs. (8) and (9). With the number of materials balances (NBAL) equal to 50 and with 1 transfer per balance (NTRPBL), the minimum number of inventory and transfer process variables per set are 52 and 54, respectively.

Each nonzero measurement error requires a separate random-number stream. Figure 5 shows 12 nonzero measurement errors for this example. Therefore, the input random-number seed file must contain at least 12 numbers. The random-number seed file containing 20 numbers is given in Fig. 6. This file of random-number seeds is used for all the examples in this manual.

The output file for this example is shown in Fig. 7. The first page consists primarily of an echo check of the input data.

```
1987987812
1897659876
1776909876
1789876545
1679456398
1987630984
1543098747
1676554679
1787656778
1389098765
1895639857
1674982654
1653987289
1673892674
1567589403
1739823648
1986540383
2098769873
1796576548
1876940846
```

Fig. 6.
Random number seeds--Example 1.

At the top of the second page is a summary of the important integers calculated in subroutine SETMAS. These integers, computed from the input process data, define the dimensions of the problem. The summary table (Sec. V) for this example appears on the bottom of the second page of the output file.

B. Example 2

This example was chosen to illustrate the case where multiple transfers take place within materials balance intervals. The block diagram is given in Fig. 8. A 1-kg input transfer takes place every hour with a 10-kg output transfer at 10-h intervals. Materials balances are calculated every 10 h. The input and output transfers and the inventory are each determined from one measured process value. These process variables are summarized in Table IX.

In this case, the input transfer occurs at a frequency of 10 transfers per balance, with the output transfer at 1 transfer per balance. However, the code requires that all transfers take place at the same frequency. This requirement is satisfied by structuring the output transfer at the same frequency as the input transfer, but with 90% of these transfers equal to zero. The transfer measurement error models are of the mixed type with the

```
INPUT DATA FILE - MESDT1
PV ARRAY FILE - PVARA1
OUTPUT DECANAL FILE - DECINN

MON, OCT 19 1981      14:56:11

TITLE FROM PROCESS MODEL CODE

EXAMPLE 1 - SAMPLE PROBLEM 1 FROM PROCESS MODEL USER'S MANUAL (LA-8761-M)

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = 0
    NUMBER OF RUNS (NRUN) =   1
    NUMBER OF BALANCES (NBAL) = 50
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) =  1
    TIME INTERVAL (DT) =   0.003
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = 0
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = 0
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = 0
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = 0
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. 0) = 0

****************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) =  8
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =  5
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      6  2  2 -5 -5

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY              PROCESS VARIABLE NUMBER
          NUMBER        (1)      (2)      (3)      (4)      (5)
            1            1        2        0        0        0
            2            3        4        0        0        0
            3            5        6        0        0        0
            4            7        0        0        0        0
            5            8        0        0        0        0

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  0

*****************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY =  8
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 1 0 0 0 0 1 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
      55  55  53  53  53  53  55  55

READING OF PROCESS VARIABLE ARRAY COMPLETE
*****************************************************

MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE

    1:  INPUT VOLUME
        INITIAL VALUE =  100.000000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.010000  0.000000       1          10000         10000

    2:  INPUT CONCENTRATION
        INITIAL VALUE =     0.050000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.010000  0.000000       1          10000         10000

    3:  TANK1 VOLUME
        INITIAL VALUE =  100.000000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.000000  0.000000       1          10000         10000

    4:  TANK1 CONCENTRATION
        INITIAL VALUE =     0.050000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.000000  0.000000       1          10000         10000

    5:  TANK2 VOLUME
        INITIAL VALUE =  100.000000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.000000  0.000000       1          10000         10000

    6:  TANK2 CONCENTRATION
        INITIAL VALUE =     0.050000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.000000  0.000000       1          10000         10000

    7:  WASTE PU
        INITIAL VALUE =     0.010000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.010000  0.000000       1          10000         10000

    8:  PRODUCT PU
        INITIAL VALUE =     4.990000
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)      INTCAL(2)
               0.010000  0.010000  0.000000       1          10000         10000
```

Fig. 7.
Output file--Example 1.

```
****************************************************************

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

    NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) =  8
    NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) =  4
    NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  4
    NUMBER OF INVENTORY COMPONENTS (NCI) =  2
    NUMBER OF TRANSFER COMPONENTS (NCT) =  3
    NUMBER OF INVENTORY MEASUREMENTS (NMI) =  2
    NUMBER OF TRANSFER MEASUREMENTS (NMT) =  2
    NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
    NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  1
    NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) = 12
    NUMBER OF PULSE COLUMNS (NCOLUM) =  0

****************************************************************

INITIAL RANDOM NUMBER SEEDS
 1042839332   273785636  1973287924  1291185537  2063904014   164960264   491400187  1352537143   271703562  1141280589
  763526641   225695806

****************************************************************

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

****************************************************************
```

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.0000E 01	1.0000E-02	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-9.5367E-07	3.4980E-02
2	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.9073E-06	6.4940E-02
3	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.8610E-06	1.0988E-01
4	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.8147E-06	1.6980E-01
5	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.7684E-06	2.4470E-01
6	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-5.7220E-06	3.3458E-01
7	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-6.6757E-06	4.3944E-01
8	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-7.6294E-06	5.5928E-01
9	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-8.5831E-06	6.9410E-01
10	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-9.5367E-06	8.4390E-01
11	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.0490E-05	1.0087E 00
12	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.1444E-05	1.1884E 00
13	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.2398E-05	1.3832E 00
14	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.3351E-05	1.5929E 00
15	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.4305E-05	1.8176E 00
16	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.5259E-05	2.0573E 00
17	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.6212E-05	2.3119E 00
18	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.7166E-05	2.5816E 00
19	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.8120E-05	2.8662E 00
20	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-1.9073E-05	3.1658E 00
21	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.0027E-05	3.4804E 00
22	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.0981E-05	3.8099E 00
23	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.1935E-05	4.1545E 00
24	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.2888E-05	4.5140E 00
25	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.3842E-05	4.8885E 00
26	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.4796E-05	5.2780E 00
27	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.5749E-05	5.6824E 00
28	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.6703E-05	6.1019E 00
29	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.7657E-05	6.5363E 00
30	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4900E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.8610E-05	6.9857E 00
31	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-2.9564E-05	7.4501E 00
32	1.0000E 01	1.0000E-02	0.0900E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.0518E-05	7.9294E 00
33	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4900E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.1471E-05	8.4238E 00
34	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.2425E-05	8.9331E 00
35	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.3379E-05	9.4574E 00
36	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.4332E-05	9.9966E 00
37	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.5286E-05	1.0551E 01
38	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.6240E-05	1.1120E 01
39	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.7193E-05	1.1704E 01
40	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.8147E-05	1.2304E 01
41	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-3.9101E-05	1.2918E 01
42	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.0054E-05	1.3547E 01
43	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.1008E-05	1.4191E 01
44	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.1962E-05	1.4850E 01
45	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.2915E-05	1.5524E 01
46	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.3869E-05	1.6213E 01
47	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.4823E-05	1.6917E 01
48	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.5776E-05	1.7636E 01
49	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.6730E-05	1.8370E 01
50	1.0000E 01	1.0000E-02	0.0000E-01	-9.5367E-07	1.4980E-02	7.4900E-03	-9.5367E-07	3.4980E-02	-4.7684E-05	1.9119E 01

Fig. 7. (cont)

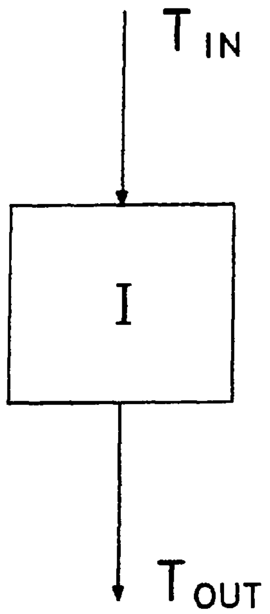

Fig. 8.
Process block diagram--Example 2.

TABLE IX

PROCESS VARIABLES FOR EXAMPLE 2

No.	Variable	Nominal Value (kg)
1	Input transfer	1
2	Inventory	100
3	Output transfer	10

uncorrelated and short-term correlated errors multiplicative and the long-term correlated errors additive. The short-term correlated errors are recalibrated every 20 h.

The input data for this example is shown in Fig. 9. At line 11, the number of runs (NRUN) is set to 100, triggering the Monte Carlo option where comparisons are made between sample and propagated CUSUM variances. To allow for multiple transfers per materials balance, NTRPBL in line 13 is set equal to 10. This example has three process variables, one for each transfer and inventory. The transfer measurement errors are set in lines 39 and 43. Uncorrelated, short-term correlated, and long-term correlated transfer errors are seen to have standard deviations of 0.1, 0.05, and 0.01, respectively. The mixed measurement error model is achieved by setting the fourth entry in lines 39 and 43 to 3. A 20 as the fifth entry on each of these lines indicates that the short-term correlated transfer errors are recalibrated every 20 transfers or every 2 materials balances.

```
/ ******* MEASIM INPUT DATA  -  EXAMPLE 2  -  SEPTEMBER 1981
/
/ ******* PARAMETERS (NBALMX=1Ø5, NTRNMX=515, NPVMX=12, NPVIMX=1Ø, NPVTMX=8,
/                     NCIMX=1Ø, NCTMX=8, NMIMX=2, NMTMX=2, NSIMX=2, NSTMX=2,
/                     NCMX=2, MXPMX=2, NSTRMX=2Ø, NCOLMX=2)
/
EXAMPLE 2 - SEPTEMBER 1981 - TEN INPUT TRANSFERS PER MATERIALS BALANCE
/
1/             IZE    - ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE)
Ø/             IRNSCH - RANDOM NUMBER SEEDS CHANGE FROM RUN TO RUN WHEN NONZERO
1ØØ/           NRUN   - NUMBER OF RUNS
5Ø/            NBAL   - NUMBER OF MATERIALS BALANCES
1Ø/            NTRPBL - NUMBER OF TRANSFERS PER BALANCE
Ø/             DT     - TIME CONSTANT (NOT USED IN THIS EXAMPLE)
Ø/             MASPRT - PRINTS MASSAGE DEBUGG OUTPUT WHEN NONZERO
Ø/             ITIPRP - PRINTS TRANSFER-INVENTORY SET NO. AND PROC. VAR. NO. IF=Ø
Ø/             IMESPR - PRINTS INPUT MEASUREMENT ERRORS WHEN EQUAL TO 1
Ø/             IPVPRT - PRINTS PROCESS VARIABLE FILE WHEN NON ZERO
Ø/             ICLAPS - REDUCES DIMENSIONS OF DECANAL INPUT FILE WHEN NONZERO
/
/ ********** INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA
/
3/             NPV    - NUMBER OF PROCESS VARIABLES
3/             NTRIN  - NUMBER OF TRANSFER AND INVENTORY SETS IN THE PROCESS
/
/   ITIN - INVENTORY-TRANSFER NUMBERS
5 1 -5/
/
/   IPVNO - PROCESS VARIABLE NUMBER CORRESPONDING TO EACH INVENTORY OR TRANSFER
1/            INPUT TRANSFER
2/            INVENTORY
3/            OUTPUT TRANSFER
/
Ø/             ISPNTI - SPECIFIC INVENTORY-TRANSFER SET NO. (Ø GIVES ALL SETS)
/
/ *********** MEASUREMENT ERRORS ASSOCIATED WITH EACH PROCESS VARIABLE
/
1:  INPUT TRANSFER
     .1 .Ø5 .Ø1 3 2Ø/
2:  INVENTORY
     .1/
3:  OUTPUT TRANSFER
     .1 .Ø5 .Ø1 3 2Ø/
```

Fig. 9.
Input data--Example 2.

In this case, the process variable file must contain many more transfers per set than inventories because there are 10 transfers for every materials balance interval. With the number of materials balances (NBAL) at 50 and the number of transfers per balance (NTRPBL) at 10, it follows from Eqs. (8) and (9) that the minimum number of inventory and transfer process variable values per set will be 52 and 513, respectively. Although the output transfer has only one transfer per materials balance, all transfers must have the same frequency. Thus, the output transfer consists of 9 zero and 1 nonzero entries for every balance period.

The output file is shown in Fig. 10. Again, it is important to note that the integers calculated in subroutine SETMAS are consistent with the process model and measurement errors. A total of 7 random-number streams is required because there are 7 nonzero measurement errors. The tabular summary of the zero error case

```
INPUT DATA FILE - MESDT6
PV ARRAY FILE - PVARA6
OUTPUT DECANAL FILE - DECINN

MON, OCT 19 1981      15:18:03

TITLE FOR PROCESS VARIABLE ARRAY

EXAMPLE 2 - SEPTEMBER 1981 - TFN INPUT TRANSFERS PER MATERIALS BALANCE

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = 0
    NUMBER OF RUNS (NRUN) = 100
    NUMBER OF BALANCES (NBAL) = 50
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) = 10
    TIME INTERVAL (DT) =   0.000
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = 0
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = 0
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = 0
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = 0
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. 0) = 0

****************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) =  3
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =  3
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      5  1 -5

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY              PROCESS VARIABLE NUMBER
          NUMBER        (1)      (2)      (3)      (4)      (5)
             1           1        0        0        0        0
             2           2        0        0        0        0
             3           3        0        0        0        0

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  0

****************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY =  3
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 0 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
     515  55 515

READING OF PROCESS VARIABLE ARRAY COMPLETE
****************************************************

MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE

   1:  INPUT TRANSFER
       INITIAL VALUE =     1.000000
               SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)        INTCAL(2)
             0.100000  0.050000  0.010000       3            20         10000

   2:  INVENTORY
       INITIAL VALUE =  100.000000
               SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)        INTCAL(2)
             0.100000  0.000000  0.000000       1          10000        10000

   3:  OUTPUT TRANSFER
       INITIAL VALUE =     0.000000
               SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)        INTCAL(2)
             0.100000  0.050000  0.010000       3            20         10000

****************************************************

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

   NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) =  3
   NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) =  1
   NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  2
   NUMBER OF INVENTORY COMPONENTS (NCI) =  1
   NUMBER OF TRANSFER COMPONENTS (NCT) =  2
   NUMBER OF INVENTORY MEASUREMENTS (NMI) =  1
   NUMBER OF TRANSFER MEASUREMENTS (NMT) =  1
   NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
   NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  2
   NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) =  7
   NUMBER OF PULSE COLUMNS (NCOLUM) =  0

****************************************************

INITIAL RANDOM NUMBER SEEDS
 1042839332   273785636  1973287924  1291185537  2063904014   164960264   491400187
```

Fig. 10.
Output file--Example 2.

****

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.0161E 02
2	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.0424E 02
3	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.0589E 02
4	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.0856E 02
5	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.1025E 02
6	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.1296E 02
7	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.1469E 02
8	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.1745E 02
9	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.1922E 02
10	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.2201E 02
11	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.2382E 02
12	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.2665E 02
13	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.2851E 02
14	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.3138E 02
15	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.3327E 02
16	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.3618E 02
17	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.3812E 02
18	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.4107E 02
19	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.4304E 02
20	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	1.0000E-04	0.0000E-01	2.0161E 02	0.0000E-01	2.4604E 02
21	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.4805E 02
22	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.5109E 02
23	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.5314E 02
24	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.5621E 02
25	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.5831E 02
26	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.6142E 02
27	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.6356E 02
28	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.6671E 02
29	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.6889E 02
30	1.0000E 02	1.0000E 02	0.0000E-01	0.9000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.7208E 02
31	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.7430E 02
32	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.7754E 02
33	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.7979E 02
34	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.8307E 02
35	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.8536E 02
36	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.8868E 02
37	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.9102E 02
38	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.9438E 02
39	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	2.9675E 02
40	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	1.0000E-04	0.0000E-01	2.0161E 02	0.0000E-01	3.0015E 02
41	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.0257E 02
42	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.0601E 02
43	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.0846E 02
44	1.0000E 02	1.0000E 02	0.0006E-01	0.0000E-01	1.6131E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.1194E 02
45	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.1444E 02
46	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.1796E 02
47	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.5101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.2050E 02
48	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.2406E 02
49	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.2664E 02
50	1.0000E 02	1.0000E 02	0.0000E-01	0.0000E-01	1.6101E 00	2.6000E-03	0.0000E-01	2.0161E 02	0.0000E-01	3.3024E 02

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-5.953200D-01	201.610	228.185	1.13181	OK
2	4.105980D-01	204.240	221.576	1.08488	OK
3	-1.291070D 00	205.890	222.249	1.07945	OK
4	3.573395D-01	208.561	262.815	1.26013	OK
5	1.842702D 00	210.252	251.656	1.19692	OK
6	6.418446D-01	212.963	312.050	1.46520	***** RATIO OUTSIDE INTERVAL
7	2.970470D-01	214.694	261.898	1.21987	OK
8	-3.833704D-01	217.446	217.763	1.00146	OK
9	1.465298D 00	219.217	264.450	1.20634	OK
10	7.950744D-01	222.009	248.874	1.12101	OK
11	1.586324D-01	223.821	288.293	1.28805	OK
12	8.916961D-01	226.653	257.636	1.13670	OK
13	3.045162D-01	228.505	234.551	1.02646	OK
14	4.147478D 00	231.378	266.092	1.15003	OK
15	1.596286D 00	233.271	253.491	1.08668	OK
16	1.173356D 00	236.184	271.735	1.15052	OK
17	8.084298D-01	238.117	244.115	1.02519	OK
18	2.455346D 00	241.070	251.560	1.04351	OK
19	7.980220D-01	243.044	267.295	1.09978	OK
20	2.313276D-02	246.037	282.811	1.14947	OK
21	3.708987D-01	248.051	289.917	1.16878	OK
22	1.132591D 00	251.085	230.949	0.919802	OK
23	1.132678D 00	253.139	304.132	1.20144	OK
24	3.068175D 00	256.214	296.293	1.15643	OK
25	2.479944D 00	258.308	226.348	0.876260	OK
26	1.283651D 00	261.423	281.308	1.07606	OK
27	2.896128D 00	263.558	334.159	1.26788	OK

Fig. 10. (cont)

28	8.040979D-01	266.713	319.013	1.19609	OK
29	3.781153D 00	268.888	299.307	1.11313	OK
30	1.832231D 00	272.084	270.789	0.995241	OK
31	1.379149D 00	274.300	229.888	0.838089	OK
32	7.003021D-01	277.535	284.004	1.02331	OK
33	2.594000D 00	279.792	300.179	1.07287	OK
34	1.505929D 00	283.068	302.610	1.06904	OK
35	2.257806D 00	285.365	321.154	1.12542	OK
36	1.246385D 00	288.682	313.690	1.08663	OK
37	6.317782D-01	291.018	306.523	1.05328	OK
38	1.813944D 00	294.375	343.788	1.16786	OK
39	1.467077D 00	296.753	371.220	1.25094	OK
40	3.867055D 00	300.151	276.636	0.921656	OK
41	1.507791D 00	302.568	325.207	1.07482	OK
42	2.414058D 00	306.006	336.094	1.09832	OK
43	2.463362D 00	308.464	345.812	1.12108	OK
44	3.799634D 00	311.943	413.736	1.32632	***** RATIO OUTSIDE INTERVAL
45	2.595830D 00	314.441	346.919	1.10329	OK
46	3.869693D 00	317.960	348.115	1.09484	OK
47	3.981794D 00	320.199	291.746	0.910286	OK
48	1.209228D 00	324.058	370.565	1.14351	OK
49	2.637879D 00	326.637	401.098	1.22796	OK
50	3.333126D 00	330.237	379.766	1.14998	OK

Fig. 10. (cont)

is given on p. 2 of Fig. 10 and the results of the 100 sample Monte Carlo run follow. The limits for the 95% confidence interval are also given. From the table, it can be seen that at materials balances 6 and 44 the ratio of the sampled-to-propagated variances lies outside the 95% confidence interval. This is a normal situation and does not necessarily suggest errors in the variance calculations.

C. Example 3

This example demonstrates the use of pulse-column inventories and transfers computed from the product of flow rate and concentration. A block diagram for this process is given in Fig. 11. This small UPAA represents a portion of the Plutonium Purification Process (PPP) from the Allied-General Nuclear Services (AGNS) reprocessing plant in Barnwell, South Carolina. It is assumed that flow rate and concentration measurements are taken every hour and materials balance calculations every 10 h.

The process variables required for this example are summarized in Table X. Because 5 process variables are required for each column inventory and 2 process variables for each transfer, it would be logical to assume that 14 process variables would be required to model this process. However, three of the process variables are used twice. The 2A column-feed concentration is used to calculate both the input transfer and the 2A column inventory. Similarly, the 2B column-product concentration is used to

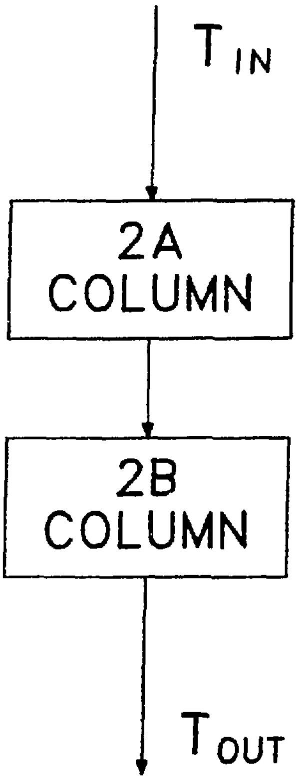

Fig. 11.
Process block diagram--Example 3.

TABLE X

PROCESS VARIABLES FOR EXAMPLE 3

No.	Variable	Nominal Value
1	Flow rate-2A feed (L/h)	106.0
2	Concentration-2A feed (kg/L)	0.0542
3	Concentration-2A waste (kg/L)	0.00004
4	Concentration-2A product (kg/L)	0.0488
5	Top organic volume-2A (L)	97.0
6	Bottom aqueous volume-2A (L)	190.0
7	Concentration-2B waste (kg/L)	0.0006
8	Top organic volume-2B (L)	81.5
9	Bottom aqueous volume-2B (L)	133.0
10	Flow rate-2B product (L/h)	160.0
11	Concentration-2B product (kg/L)	0.0359

calculate both the 2B column inventory and the output transfer. The product concentration for the 2A column is identical to the feed concentration for the 2B column. Therefore, this process variable is used in inventory calculations for both columns. In those cases where a process variable is used for both transfer and inventory calculations, the process variable should be considered as a transfer in the process variable file to assure sufficient transfer values to perform the materials balance calculations. The number of transfer values per set must always be equal to or greater than the number of inventory values per set.

The flow rates and concentrations associated with the input and output transfers have nonzero uncorrelated and short-term correlated errors and zero long-term correlated errors. Flow-rate measurements are recalibrated every 20 h. No other measurements are recalibrated. As usual, inventory correlated errors are assumed zero because these errors tend to cancel in the materials balance variance calculations.

The use of the same process variable in the transfer and inventory leads to correlations between transfers and inventories. The input transfer is correlated with 2A column inventory through the 2A feed concentration, and the output transfer is correlated with the 2B column inventory through the 2B product concentration. In addition, the 2A and 2B column inventories are correlated through the 2A product concentration because both columns use this variable for inventory calculations. The MEASIM code has not been designed to handle the correlations among transfers and inventories and the correlations among different inventories. Under steady-state conditions, these additional correlations will have a relatively small contribution to the materials balance variance because the initial and final inventories appear with opposite signs in the materials balance equation (see Appendix A). This theory is confirmed through the results of a Monte Carlo run discussed below.

The input data for this example are given in Fig. 12. At line 11 the number of runs (NRUN) is set at 100 to indicate a Monte Carlo simulation. The number of materials balances (NBAL)

```
/ ******** COLUMN 2A AND COLUMN 2B USING OPERATING CONDITIONS FROM MINIRUN #6
/
/ ******* PARAMETERS (NBALMX=1Ø5, NTRNMX=515, NPVMX=12, NPVIMX=1Ø, NPVTMX=8,
/                      NCIMX=1Ø, NCTMX=8, NMIMX=2, NMTMX=2, NSIMX=2, NSTMX=2,
/                      NCMX=2, MXPMX=2, NSTRMX=2Ø, NCOLMX=2)
/
COLUMNS 2A AND 2B USING OPERATING CONDITIONS FROM AGNS MINIRUN #6
/
1/           IZE    - ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE)
Ø/           IRNSCH - RANDOM NUMBER SEEDS CHANGE FROM RUN TO RUN WHEN NONZERO
1ØØ/         NRUN   - NUMBER OF RUNS
1Ø/          NBAL   - NUMBER OF MATERIALS BALANCES
1Ø/          NTRPBL - NUMBER OF TRANSFERS PER BALANCE
1./          DT     - TIME INTERVAL (DT)
Ø/           MASPRT - PRINTS MASSAGE DEBUGG OUTPUT WHEN NONZERO
Ø/           ITIPRP - PRINTS TRANSFER-INVENTORY SET NO. AND PROC. VAR. NO. IF=Ø
Ø/           IMESPR - PRINTS INPUT MEASUREMENT ERRORS WHEN EQUAL TO 1
Ø/           IPVPRT - PRINTS PROCESS VARIABLE FILE WHEN NON ZERO
2/           ICLAPS - COLLAPSES MATRIX OUTPUT TO SCALAR FORM WHEN SET TO 1
/
/ ********** INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA
/
11/          NPV    - NUMBER OF PROCESS VARIABLES
4/           NTRIN  - NUMBER OF TRANSFER AND INVENTORY SETS IN THE PROCESS
/
/   ITIN - INVENTORY-TRANSFER NUMBERS
7 3 4 -7/
/
/   IPVNO - PROCESS VARIABLE NUMBER CORRESPONDING TO EACH INVENTORY OR TRANSFER
1 2/           INPUT TRANSFER
2 3 4 5 6/   2A COLUMN INVENTORY
4 7 11 8 9/ 2B COLUMN INVENTORY
1Ø 11/         OUTPUT TRANSFER
/
Ø/           ISPNTI - SPECIFIC INVENTORY-TRANSFER SET NO. (Ø GIVES ALL SETS)
/
/ ************ MEASUREMENT ERRORS ASSOCIATED WITH EACH PROCESS VARIABLE
/
1:  FLOW RATE - 2AF
    .Ø2 .Ø2 Ø. 1 2Ø/
2:  CONC. - 2AF
    .Ø64397 .1418/
3:  CONC. - 2AW
    .1/
4:  CONC. - 2AP
    .Ø25331/
5:  TOP VOLUME
    Ø/
6:  BOTTOM VOLUME
    Ø/
7:  CONC. - 2BW
    .1/
8: TOP VOLUME
    Ø/
9: BOTTOM VOLUME
    Ø/
1Ø: FLOW RATE - 2BP
    .Ø2 .Ø2 Ø. 1 2Ø/
11: CONC. - 2BP
    .Ø6 .Ø1/
/
/ ******  COLUMN DATA  - (NCC(I,J),I=1,5), (VCC(I,J),I=1,2), (CCC(I,J),I=1,3)
/                ALL TEN VALUES ON ONE LINE  (I.E. ONE LINE PER COLUMN)
/
.25 3.64 5.82 .45 9.63 97 19Ø .Ø6 1.E-3 .Ø6 /  COLUMN 2A
.72 1.62 3.26 Ø Ø 81.5 133 .Ø4 1.E-3 .Ø6 /   COLUMN 2B
1
2
3
4
```

Fig. 12.
Input data--Example 3.

is set to 10 with the number of transfers per balance (NTRPBL) equal to 10. Because the time interval between flow-rate measurements is 1 h, DT is set to 1.0 at line 14. Thus, with each transfer separated by 1 h and with 10 transfers per materials balance, the materials balance calculations will be made at 10-h intervals. The actual process variables for each of the four transfers and inventories are selected in input data lines 30-33, which show that process variables 2, 4, and 11 are each used twice. The "20" appearing as the fifth entry in lines 40 and 58 instructs the code to recalibrate the flow-rate measurements every 20th balance. The appearance of 1, 2, 3, and 4 in the last four lines, respectively, of the input file instructs the code to make isolated runs for each of the four transfers and inventories associated with the UPAA.

The output file for this example is shown in Fig. 13. The first page is devoted to the echo check of input data. A summary of the subroutine SETMAS calculations on p. 2 indicates that the number of inventory and transfer process variables are 10 and 4, respectively. This number may seem somewhat contradictory because there are only a total of 11 process variables. Though three of the process variables assume dual roles as discussed above, the calculations in subroutine SETMAS do not take this into consideration but simply allocate five process variables for each column and two process variables for each transfer. Normally each inventory element, such as a tank or container, has one associated inventory component. Each pulsed column has three inventory components. A fourth inventory component is required to accumulate the constant terms for all the pulsed columns. Thus, the first pulsed column requires four inventory components and each succeeding column requires only three inventory components. This example has two pulsed columns, resulting in a total of seven (4 + 3) inventory components. Each transfer computed from the product of flow rate and concentration requires four transfer components. Because both the input and output transfers are computed in this way, the total number of transfer components will be equal to 8.

```
INPUT DATA FILE - MESDT8
PV ARRAY FILE - PVARA8
OUTPUT DECANAL FILE - DECINN

MON, OCT 19 1981      15:44:32

TITLE FOR PROCESS VARIABLE ARRAY

COLUMNS 2A AND 2B USING OPERATING CONDITIONS FROM AGNS MINIRUN #6

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = Ø
    NUMBER OF RUNS (NRUN) = 1ØØ
    NUMBER OF BALANCES (NBAL) = 1Ø
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) = 1Ø
    TIME INTERVAL (DT) =    1.ØØØ
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = Ø
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = Ø
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = Ø
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = Ø
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. Ø) = 2

****************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) = 11
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =   4
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      7  3  4 -7

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY             PROCESS VARIABLE NUMBER
          NUMBER       (1)     (2)     (3)     (4)     (5)
            1           1       2       Ø       Ø       Ø
            2           2       3       4       5       6
            3           4       7      11       8       9
            4          1Ø      11       Ø       Ø       Ø

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  Ø

****************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY = 11
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 1 Ø Ø Ø Ø Ø Ø Ø 1 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
     115 115  12  12  12  12  12  12  12 115 115

READING OF PROCESS VARIABLE ARRAY COMPLETE
****************************************************

MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE

    1:  FLOW RATE - 2AF
        INITIAL VALUE =  106.ØØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.Ø2ØØØØ  Ø.Ø2ØØØØ   Ø.ØØØØØØ      1            2Ø       1ØØØØ
    2:  CONC. - 2AF
        INITIAL VALUE =     Ø.Ø542ØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.Ø64397  Ø.141ØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    3:  CONC. - 2AW
        INITIAL VALUE =     Ø.ØØØØ4Ø
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.1ØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    4:  CONC. - 2AP
        INITIAL VALUE =     Ø.Ø488ØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.Ø25331  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    5:  TOP VOLUME
        INITIAL VALUE =    97.ØØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.ØØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    6:  BOTTOM VOLUME
        INITIAL VALUE =  19Ø.ØØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.ØØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    7:  CONC. - 2BW
        INITIAL VALUE =     Ø.ØØØ6ØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.1ØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    8: TOP VOLUME
        INITIAL VALUE =    81.5ØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.ØØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
    9: BOTTOM VOLUME
        INITIAL VALUE =  133.ØØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.ØØØØØØ  Ø.ØØØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
   1Ø: FLOW RATE - 2BP
        INITIAL VALUE =  16Ø.ØØØØØØ
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.Ø2ØØØØ  Ø.Ø2ØØØØ   Ø.ØØØØØØ      1            2Ø       1ØØØØ
   11: CONC. - 2BP
        INITIAL VALUE =     Ø.Ø3587Ø
                 SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP    INTCAL(1)     INTCAL(2)
               Ø.Ø6ØØØØ  Ø.Ø1ØØØØ   Ø.ØØØØØØ      1         1ØØØØ       1ØØØØ
```

Fig. 13.
Output file--Example 3.

```
*****************************************************

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

    NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) = 10
    NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) = 10
    NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  4
    NUMBER OF INVENTORY COMPONENTS (NCI) =  7
    NUMBER OF TRANSFER COMPONENTS (NCT) =  8
    NUMBER OF INVENTORY MEASUREMENTS (NMI) =  1
    NUMBER OF TRANSFER MEASUREMENTS (NMT) =  2
    NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
    NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  1
    NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) = 16
    NUMBER OF PULSE COLUMNS (NCOLUM) =  2

*****************************************************

COLUMN CONSTANTS (HCC(5,I), VCC(3,I), CCC(3,I)
 0.250000E 00  0.364000E 01  0.582000E 01  0.450000E 00  0.963000E 01  0.970000E 02  0.190000E 03  0.600000E-01
 0.100000E-02  0.600000E-01
 0.720000E 00  0.162000E 01  0.326000E 01  0.000000E 00  0.000000E 00  0.815000E 02  0.133000E 03  0.400000E-01
 0.100000E-02  0.600000E-01

INITIAL RANDOM NUMBER SEEDS
 1042839332  273705636  1973287924  1291185537  2063904014  164960264  491400187  1352537143  271703562  1141280589
  763526641  225695806  1653987289  1673092674  1567589403  1739823648


*****************************************************

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

*****************************************************
```

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.7757E 01	2.1278E-01	9.3136E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2003E 01	6.9309E 01	5.9967E-02	7.2410E 01	5.9967E-02	7.2410E 01
2	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.6683E 01	5.9967E-02	7.2551E 01	1.1993E-01	2.8322E 02
3	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.9309E 01	5.9967E-02	7.2551E 01	1.7990E-01	6.2214E 02
4	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.6683E 01	5.9967E-02	7.2551E 01	2.3987E-01	1.0997E 03
5	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.9309E 01	5.9967E-02	7.2551E 01	2.9984E-01	1.7053E 03
6	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.6683E 01	5.9967E-02	7.2551E 01	3.5980E-01	2.4496E 03
7	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.9309E 01	5.9967E-02	7.2551E 01	4.1977E-01	3.3219E 03
8	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.6683E 01	5.9967E-02	7.2551E 01	4.7974E-01	4.3328E 03
9	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.9309E 01	5.9967E-02	7.2551E 01	5.3970E-01	5.4710E 03
10	1.7757E 01	2.1278E-01	9.3136E-03	5.9967E-02	7.2144E 01	6.6683E 01	5.9967E-02	7.2551E 01	5.9967E-01	6.7495E 03

```
*****************************************************

RESULTS OF MONTE CARLO SIMULATION WITH   100 SAMPLES

*****************************************************

CHI-SQUARE/.N-1) RATIO FOR 95% CONFIDENCE
    UPPER LIMIT =    1.29223
    LOWER LIMIT =  0.736473
```

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-3.228776D-01	72.4101	76.4173	1.05534	OK
2	-7.734398D-01	283.220	293.831	1.03747	OK
3	-9.624837D-01	622.139	662.270	1.06451	OK
4	-1.321164D 00	1099.69	1217.84	1.10744	OK
5	-2.207251D 00	1705.34	1895.49	1.11150	OK
6	-2.875400D 00	2449.56	2716.74	1.10907	OK
7	-2.630043D 00	3321.89	3662.20	1.10244	OK
8	-3.172722D 00	4332.82	4755.76	1.09761	OK
9	-3.685706D 00	5471.85	6033.34	1.10261	OK
10	-4.283620D 00	6749.48	7505.24	1.11197	OK

```
*****************************************************

SUMMARY FOR INVENTORY AND TRANSFER NUMBER = 1

*****************************************************
```

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9111E 01	6.7682E 01	5.7452E 01	6.9111E 01	5.7452E 01	6.9111E 01
2	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.6363E 01	5.7452E 01	6.9186E 01	1.1490E 02	2.7367E 02
3	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.7682E 01	5.7452E 01	6.9186E 01	1.7236E 02	6.0833E 02
4	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.6363E 01	5.7452E 01	6.9186E 01	2.2981E 02	1.0784E 03
5	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.7682E 01	5.7452E 01	6.9186E 01	2.8726E 02	1.6785E 03
6	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.6363E 01	5.7452E 01	6.9186E 01	3.4471E 02	2.4139E 03
7	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.7682E 01	5.7452E 01	6.9186E 01	4.0216E 02	3.2795E 03
8	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.6363E 01	5.7452E 01	6.9186E 01	4.5962E 02	4.2804E 03
9	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.7682E 01	5.7452E 01	6.9186E 01	5.1707E 02	5.4114E 03
10	0.0000E-01	0.0000E-01	0.0000E-01	5.7452E 01	6.9186E 01	6.6363E 01	5.7452E 01	6.9186E 01	5.7452E 02	6.6778E 03

Fig. 13. (cont)

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	5.667958D 01	69.1114	71.9254	1.04072	OK
2	1.133847D 02	273.674	286.406	1.04652	OK
3	1.703292D 02	608.329	657.748	1.08124	OK
4	2.272259D 02	1078.35	1190.13	1.10366	OK
5	2.838518D 02	1678.46	1848.29	1.10118	OK
6	3.407061D 02	2413.94	2645.49	1.09592	OK
7	3.976738D 02	3279.50	3598.18	1.09717	OK
8	4.545408D 02	4280.42	4707.11	1.09969	OK
9	5.115601D 02	5411.41	5959.34	1.10125	OK
10	5.683322D 02	6677.76	7328.02	1.09738	OK

****

SUMMARY FOR INVENTORY AND TRANSFER NUMBER = 2

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
2	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
3	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
4	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
5	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
6	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
7	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
8	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
9	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01
10	1.6163E 01	1.6451E-01	8.0398E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	3.1294E-01	0.0000E-01	3.1294E-01

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-4.974606D-02	0.312945	0.284004	0.907520	OK
2	3.474834D-02	0.312945	0.393534	1.25752	OK
3	-5.576336D-02	0.312945	0.248648	0.794543	OK
4	2.612707D-02	0.312945	0.268062	0.856578	OK
5	2.490187D-02	0.312945	0.293676	0.938429	OK
6	3.764761D-02	0.312945	0.222961	0.712462	***** RATIO OUTSIDE INTERVAL
7	-5.584248D-03	0.312945	0.342351	1.09397	OK
8	-9.881632D-02	0.312945	0.393477	1.25734	OK
9	-2.278257D-02	0.312945	0.285609	0.912650	OK
10	-1.423687D-02	0.312945	0.334486	1.06883	OK

****

SUMMARY FOR INVENTORY AND TRANSFER NUMBER = 3

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
2	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
3	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
4	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
5	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
6	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
7	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
8	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
9	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02
10	1.5946E 00	4.8270E-02	1.2738E-03	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	9.3993E-02	0.0000E-01	9.3993E-02

Fig. 13. (cont)

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-2.322830D-02	9.399260E-02	8.262150E-02	0.879021	OK
2	2.453317D-02	9.399260E-02	9.903030E-02	1.05360	OK
3	-1.925673D-02	9.399260E-02	0.116360	1.23797	OK
4	1.076840D-02	9.399260E-02	8.633649E-02	0.918545	OK
5	-5.620098D-03	9.399260E-02	8.706079E-02	0.926251	OK
6	-2.007064D-02	9.399260E-02	0.102916	1.09494	OK
7	7.991161D-03	9.399260E-02	8.523972E-02	0.906876	OK
8	-9.824753D-03	9.399260E-02	8.036880E-02	0.855054	OK
9	-1.636000D-02	9.399260E-02	0.101376	1.07856	OK
10	8.608046D-03	9.399260E-02	9.020430E-02	0.959695	OK

****

SUMMARY FOR INVENTORY AND TRANSFER NUMBER = 4

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.8985E 00	1.6468E 00	-5.7392E 01	2.8985E 00	-5.7392E 01	2.8985E 00
2	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	3.2938E-01	-5.7392E 01	2.9644E 00	-1.1478E 02	9.1547E 00
3	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	1.6468E 00	-5.7392E 01	2.9644E 00	-1.7218E 02	1.3439E 01
4	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	3.2938E-01	-5.7392E 01	2.9644E 00	-2.2957E 02	2.1014E 01
5	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	1.6468E 00	-5.7392E 01	2.9644E 00	-2.8696E 02	2.6614E 01
6	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	3.2938E-01	-5.7392E 01	2.9644E 00	-3.4435E 02	3.5507E 01
7	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	1.6468E 00	-5.7392E 01	2.9644E 00	-4.0174E 02	4.2474E 01
8	0.0000E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	3.2938E-01	-5.7392E 01	2.9644E 00	-4.5914E 02	5.7614E 01
9	0.0006E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	1.6468E 00	-5.7392E 01	2.9644E 00	-5.1653E 02	6.0869E 01
10	0.0006E-01	0.0000E-01	0.0000E-01	-5.7392E 01	2.9644E 00	3.2938E-01	-5.7392E 01	2.9644E 00	-5.7392E 02	7.2397E 01

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-5.711736D 01	2.89853	3.25158	1.12180	OK
2	-1.142507D 02	9.15674	9.86271	1.07710	OK
3	-1.715814D 02	13.4387	16.6655	1.24012	OK
4	-2.288638D 02	21.0143	28.3439	1.34879	***** RATIO OUTSIDE INTERVAL
5	-2.859110D 02	26.6138	34.7483	1.30565	***** RATIO OUTSIDE INTERVAL
6	-3.431702D 02	35.5069	43.0374	1.21209	OK
7	-4.005432D 02	42.4238	50.7457	1.19616	OK
8	-4.578281D 02	52.6345	62.7123	1.19147	OK
9	-5.152344D 02	60.8689	72.9338	1.19821	OK
10	-5.724127D 02	72.3970	84.7511	1.17064	OK

Fig. 13. (cont)

The remainder provides the output summaries for the complete UPAA and for each isolated inventory and transfer set. The zero-error output table and the Monte Carlo results are given for each of these five cases. As indicated previously, the Monte Carlo runs are obtained by setting the NRUN input variable to a value greater than 1. The results of the Monte Carlo simulation on p. 2 for the complete simulation are particularly interesting.

The MEASIM code neglects correlations between transfers and inventories and between different column inventories when computing the propagated or analytical CUSUM variances. However, the sample CUSUM σ^2 calculations from the Monte Carlo simulation

must, by their very nature, include all these correlations. The ratio of the sample and propagated CUSUM variances in the table on p. 2 is within the 95% confidence interval for all 10 balances. This ratio adds substance to the assumption that the correlations mentioned above can be safely neglected in the steady-state case. The remainder of the output file summarizes the results of the simulation for each transfer and inventory element considered separately. These isolation runs prove very useful for program debugging purposes.

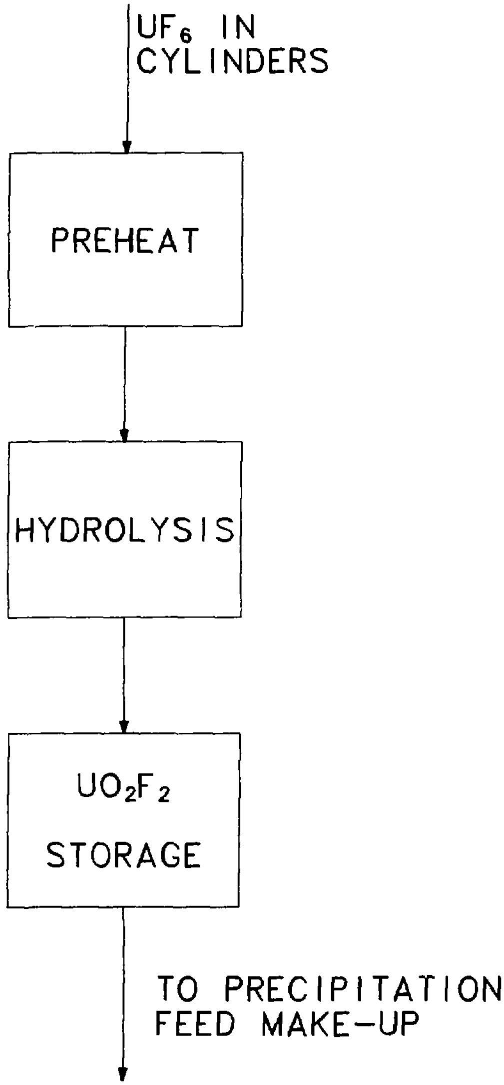

Fig. 14.
Process block diagram--Example 4.

D. Example 4

The UPAA for this example, as shown in Fig. 14, is taken from a portion of a UF_6-to-oxide conversion process. This example was chosen because it contains a transfer computed from the product of concentration and difference in weights. Also, an additive error model is used in this example. The SNM is uranium.

After the UF_6 enters the UPAA from cylinders, the amount of uranium transferred is calculated from the product of the UF_6 concentration and the difference between the full and empty weight of the cylinder; that is,

$$T_{IN} = C(W_i - W_f) \quad ,$$

where

T_{IN} = input U (kg),
C = input U concentration (kg/kg),
W_i = initial cylinder weight (kg), and
W_f = final cylinder weight (kg).

Because initial and final cylinder weights are measured on the same instrument, these measurements are correlated and are shown to have the same correlated error. The batch output transfer is determined from the product of the volume and uranium concentration in the UO_2F_2 storage tank. No inventory measurements are made in this UPAA. The cylinder weight measurement errors are additive, while all the other measurement errors are multiplicative. These batch input and output transfers take place at a frequency of two per day with the materials balance calculations performed once per day.

The process variables for this example along with their nominal values are given in Table XI. The input data are given in Fig. 15. Setting the number of runs (NRUN) equal to 100 in line 11 of the input data selects the Monte Carlo option with 100 samples. The number of transfers per balance period (NTRPBL) is set to 2 at line 13. An "8" for the first entry in line 27 for the inventory-transfer number selects the input transfer. Process variables corresponding to this input transfer are defined in line 30. An additive error model for weight measurements is selected with the "2" as the fourth entry in lines 40 and 42.

TABLE XI

PROCESS VARIABLES FOR EXAMPLE 4

No.	Variable	Nominal Value
1	Input concentration (kg/kg)	0.676
2	Initial cylinder weight (kg)	25.0
3	Final cylinder weight (kg)	0.5
4	Output volume (L)	75.0
5	Output concentration (kg/L)	0.2208

```
/ ******* EXAMPLE 4 - INPUT TRANSFER AS PRODUCT OF CONC. AND WT. DIFF.
/
/ ******* PARAMETERS (NBALMX=1Ø5, NTRNMX=515, NPVMX=12, NPV1MX=1Ø, NPVTMX=8,
/                    NC1MX=1Ø, NCTMX=8, NMIMX=2, NMTMX=2, NSIMX=2, NSTMX=2,
/                    NCMX=2, MXPMX=2, NSTRMX=2Ø, NCOLMX=2)
/
EX. 4 - INPUT TRANSFER AS THE PRODUCT OF CONCENTRATION AND WEIGHT DIFFERENCE
/
1/          IZE    - ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE)
Ø/          IRNSCH - RANDOM NUMBER SEEDS CHANGE FROM RUN TO RUN WHEN NONZERO
1ØØ/        NRUN   - NUMBER OF RUNS
5Ø/         NBAL   - NUMBER OF MATERIALS BALANCES
2/          NTRPBL - NUMBER OF TRANSFERS PER BALANCE
Ø./         DT     - TIME INTERVAL (DT)
Ø/          MASPRT - PRINTS MASSAGE DEBUGG OUTPUT WHEN NONZERO
Ø/          ITIPRP - PRINTS TRANSFER-INVENTORY SET NO. AND PROC. VAR. NO. IF=Ø
Ø/          IMESPR - PRINTS INPUT MEASUREMENT ERRORS WHEN EQUAL TO I
Ø/          IPVPRT - PRINTS PROCESS VARIABLE FILE WHEN NON ZERO
2/          ICLAPS - COLLAPSES MATRIX OUTPUT TO SCALAR FORM WHEN SET TO 1
/
/ ********** INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA
/
5/          NPV    - NUMBER OF PROCESS VARIABLES
2/          NTRIN  - NUMBER OF TRANSFER AND INVENTORY SETS IN THE PROCESS
/
/   ITIN - INVENTORY-TRANSFER NUMBERS
8 -6/
/
/   IPVNO - PROCESS VARIABLE NUMBER CORRESPONDING TO EACH INVENTORY OR TRANSFER
1 2 3/     INPUT TRANSFER AS PRODUCT OF CONCENTRATION AND WEIGHT CHANGE
4 5/       OUTPUT TRANSFER AS PRODUCT OF VOLUME AND CONCENTRATION
/
Ø/          ISPNTI - SPECIFIC INVENTORY-TRANSFER SET NO. (Ø GIVES ALL SETS)
/
/ *********** MEASUREMENT ERRORS ASSOCIATED WITH EACH PROCESS VARIABLE
/
1:  INPUT CONCENTRATION
     .ØØ3 .ØØ1/
2:  FULL WEIGHT
     .ØØ2 .ØØ1 Ø 2/
3:  EMPTY WEIGHT
     .ØØ2 .ØØ1 Ø 2/
4:  OUTPUT VOLUME
     .Ø1 .ØØ5/
5:  OUTPUT CONCENTRATION
     .ØØ3 .ØØ1/
```

Fig. 15.
Input data--Example 4.

The output file for this example is given in Fig. 16. Although there are ten nonzero measurement errors, only nine random-number streams are required because the full-weight and empty-weight measurements both have the same correlated error and, therefore, use the same random-number stream. The Monte Carlo results given at the end of the output show good agreement between the sampled and propagated CUSUM variances with only one of the ratios lying outside the 95% confidence interval.

E. Example 5

The UPAA for this example shown in Fig. 17 is also part of a UF_6-to-oxide conversion process. This example was chosen because of correlations between the input transfer and inventory and also because of the slightly different approach that must be taken to model the ADU CAKE output.

```
INPUT DATA FILE - MESDT9
PV ARRAY FILE - PVARA9
OUTPUT DECANAL FILE - DECINN

TUE, OCT 2Ø 1981      Ø1:Ø7:35

TITLE FROM PROCESS VARIABLE FILE

EX. 4 - INPUT TRANSFER AS THE PRODUCT OF CONCENTRATION AND WEIGHT DIFFERENCE

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = Ø
    NUMBER OF RUNS (NRUN) = 1ØØ
    NUMBER OF BALANCES (NBAL) = 5Ø
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) =  2
    TIME INTERVAL (DT) =   Ø.ØØØ
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = Ø
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = Ø
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = Ø
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = Ø
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. Ø) = 2


*******************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) =  5
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =  2
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      8 -6

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY              PROCESS VARIABLE NUMBER
          NUMBER        (1)      (2)      (3)      (4)      (5)
             1           1        2        3        Ø        Ø
             2           4        5        Ø        Ø        Ø

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  Ø

*******************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY =  5
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 1 1 1 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
     115 115 115 115 115

READING OF PROCESS VARIABLE ARRAY COMPLETE
*******************************************************
```

Fig. 16.
Output file--Example 4.

MEASUREMENT ERRORS FOR EACH PROCESS VAR.ABLE

```
1: INPUT CONCENTRATION
   INITIAL VALUE =    0.676000
          SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP   INTCAL(1)   INTCAL(2)
        0.003000   0.001000   0.000000      1        10000       10000

2: FULL WEIGHT
   INITIAL VALUE =   25.000000
          SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP   INTCAL(1)   INTCAL(2)
        0.002000   0.001000   0.000000      2        10000       10000

3: EMPTY WEIGHT
   INITIAL VALUE =    0.500000
          SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP   INTCAL(1)   INTCAL(2)
        0.002000   0.001000   0.000000      2        10000       10000

4: OUTPUT VOLUME
   INITIAL VALUE =   75.000000
          SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP   INTCAL(1)   INTCAL(2)
        0.010000   0.005000   0.000000      1        10000       10000

5: OUTPUT CONCENTRATION
   INITIAL VALUE =    0.220827
          SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP   INTCAL(1)   INTCAL(2)
        0.003000   0.001000   0.000000      1        10000       10000
```

****

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

```
NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) =  5
NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) =  0
NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  5
NUMBER OF INVENTORY COMPONENTS (NCI) =  0
NUMBER OF TRANSFER COMPONENTS (NCT) =  3
NUMBER OF INVENTORY MEASUREMENTS (NMI) =  1
NUMBER OF TRANSFER MEASUREMENTS (NMT) =  2
NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  1
NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) =  9
NUMBER OF PULSE COLUMNS (NCOLUM) =  0
```

****

INITIAL RANDOM NUMBER SEEDS

```
1042839332  273785636  1373287924  1291185537  2063984814  164960264  491400187  1352537143  271783562
```

****

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-6.1035E-05	9.4615E-02
2	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.2207E-04	2.4057E-01
3	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.8311E-04	4.6186E-01
4	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.4414E-04	7.3450E-01
5	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-3.0518E-04	1.0665E 00
6	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-3.6621E-04	1.4578E 00
7	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-4.2725E-04	1.9084E 00
8	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-4.8828E-04	2.4184E 00
9	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-5.4932E-04	2.9878E 00
10	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-6.1035E-04	3.6164E 00
11	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-6.7139E-04	4.3044E 00
12	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-7.3242E-04	5.0518E 00
13	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-7.9346E-04	5.8585E 00
14	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-8.5449E-04	6.7245E 00
15	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-9.1553E-04	7.6498E 00
16	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-9.7656E-04	8.6345E 00
17	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.0376E-03	9.6786E 00
18	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.0986E-03	1.0782E 01
19	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.1597E-03	1.1945E 01
20	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.2207E-03	1.3167E 01
21	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.2817E-03	1.4440E 01
22	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.3428E-03	1.5789E 01
23	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.4038E-03	1.7139E 01
24	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.4648E-03	1.8648E 01
25	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.5259E-03	2.0167E 01
26	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.5869E-03	2.1745E 01
27	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.6479E-03	2.3383E 01
28	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.7090E-03	2.5079E 01
29	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.7700E-03	2.6835E 01
30	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.8311E-03	2.8651E 01
31	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.8921E-03	3.0526E 01
32	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-1.9531E-03	3.2460E 01
33	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.0142E-03	3.4453E 01
34	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.0752E-03	3.6506E 01
35	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.1362E-03	3.8618E 01
36	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.1973E-03	4.0790E 01
37	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.2583E-03	4.3020E 01
38	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.3193E-03	4.5310E 01
39	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.3804E-03	4.7660E 01
40	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.4414E-03	5.0069E 01
41	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.5024E-03	5.2537E 01
42	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.5635E-03	5.5064E 01
43	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.6245E-03	5.7651E 01
44	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.6855E-03	6.0297E 01
45	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.7466E-03	6.3043E 01
46	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.8076E-03	6.5768E 01
47	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.8687E-03	6.8592E 01
48	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.9297E-03	7.1475E 01
49	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-2.9907E-03	7.4418E 01
50	0.0000E-01	0.0000E-01	0.0000E-01	-6.1035E-05	9.4615E-02	2.9670E-02	-6.1035E-05	9.4615E-02	-3.0518E-03	7.7420E 01

Fig. 16. (cont)

****

RESULTS OF MONTE CARLO SIMULATION WITH 100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE
UPPER LIMIT = 1.29223
LOWER LIMIT = 0.736473

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	6.691038D-02	9.461546E-02	0.128338	1.35641	***** RATIO OUTSIDE INTERVAL
2	6.006325D-02	0.248570	0.315915	1.27093	OK
3	5.769838D-02	0.461864	0.543116	1.17592	OK
4	1.000306D-01	0.734498	0.832650	1.13363	OK
5	1.208112D-01	1.06647	1.25027	1.17235	OK
6	1.514596D-01	1.45778	1.63271	1.12000	OK
7	1.746360D-01	1.90843	2.08331	1.09163	OK
8	1.765508D-01	2.41842	2.45670	1.01583	OK
9	1.573342D-01	2.98775	3.16687	1.05995	OK
10	1.724583D-01	3.61642	3.72845	1.03098	OK
11	2.030805D-01	4.30443	4.29351	0.997464	OK
12	2.309371D-01	5.05177	5.16867	1.02314	OK
13	2.663410D-01	5.85845	5.85345	0.999147	OK
14	2.713704D-01	6.72448	6.50676	0.967623	OK
15	3.090993D-01	7.64984	7.42983	0.971241	OK
16	3.049111D-01	8.63454	8.41628	0.974723	OK
17	3.365220D-01	9.67858	9.39711	0.970918	OK
18	3.994539D-01	10.7820	10.3208	0.957226	OK
19	4.167498D-01	11.9447	11.4031	0.954664	OK
20	4.011782D-01	13.1667	12.6164	0.958202	OK
21	4.069777D-01	14.4481	13.7619	0.952506	OK
22	3.900737D-01	15.7888	14.9285	0.945510	OK
23	4.304157D-01	17.1889	16.5512	0.962898	OK
24	4.210445D-01	18.6483	17.8079	0.954935	OK
25	4.485571D-01	20.1670	19.2233	0.953203	OK
26	4.896123D-01	21.7451	20.9906	0.965303	OK
27	4.856675D-01	23.3825	22.6014	0.966592	OK
28	4.790300D-01	25.0793	24.3736	0.971061	OK
29	5.467979D-01	26.8354	25.9163	0.965752	OK
30	5.904708D-01	28.6508	27.7990	0.970269	OK
31	5.970294D-01	30.5256	29.5267	0.967277	OK
32	5.792130D-01	32.4597	31.0094	0.955318	OK
33	6.069369D-01	34.4532	32.7809	0.951463	OK
34	6.202999D-01	36.5060	34.8033	0.953360	OK
35	6.260624D-01	38.6181	37.0322	0.958935	OK
36	6.038017D-01	40.7895	38.7230	0.949336	OK
37	6.542608D-01	43.0203	40.5459	0.942483	OK
38	7.034967D-01	45.3104	42.2993	0.933545	OK
39	7.139121D-01	47.6599	44.7973	0.939937	OK
40	7.235217D-01	50.0687	47.6174	0.951040	OK
41	7.045538D-01	52.5368	49.9222	0.950233	OK
42	7.066565D-01	55.0643	53.0318	0.963088	OK
43	7.260264D-01	57.6511	55.5405	0.963390	OK
44	7.610530D-01	60.2973	57.7972	0.958537	OK
45	7.794856D-01	63.0028	60.0487	0.953111	OK
46	8.266530D-01	65.7676	63.0584	0.958806	OK
47	8.485909D-01	68.5917	65.9356	0.961276	OK
48	8.787024D-01	71.4752	68.6551	0.960544	OK
49	8.329083D-01	74.4181	72.3880	0.972721	OK
50	8.493900D-01	77.4202	75.0395	0.969249	OK

Fig. 16. (cont)

The input transfer to the precipitation feed makeup unit is at a rate of 12 batches per day with a volume of $\sim$12.5 L/batch. For each batch, the input volume is determined by measuring the volume in the precipitation feed makeup tank. However, this unit cannot be emptied completely between batches because of a residue of material left on the walls. Although this residue cannot be measured, it is estimated using both uncorrelated and correlated error components. The correlation between the input transfer and the holdup can be analyzed by considering the relevant terms in the UPAA materials balance equation. For a 1-day materials balance,

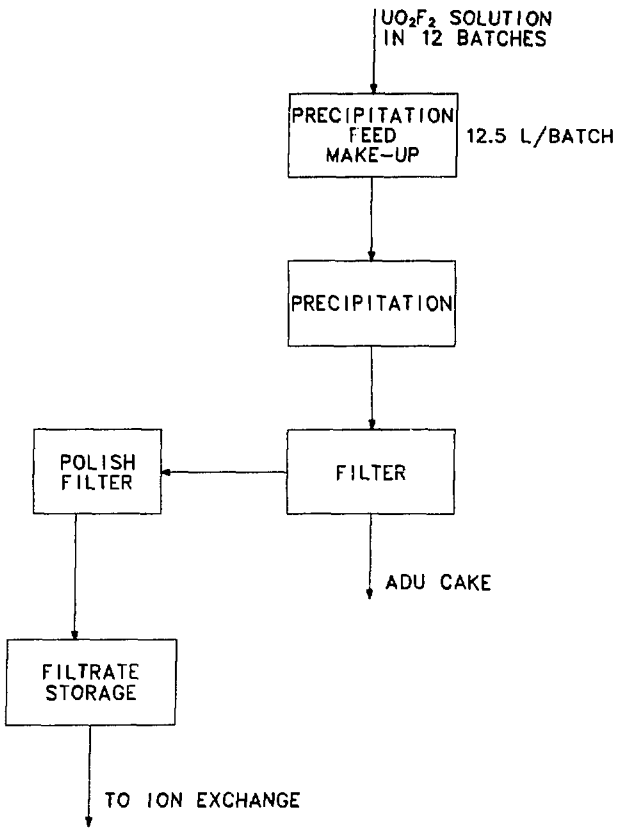

Fig. 17.
Process block diagram--
Example 5.

$$MB_1 = T_{IN} + H_0 - H_{12} \quad ,$$

where

T_{IN} =input transfer,
H_0 = initial holdup (residue), and
H_{12} =final holdup after 1 day (12 batches).

The input transfer for the 12 batches will be

$$T_{IN} = \Sigma(C_iV_i - H_{i-1}) \quad ,$$

where

Σ = summation over i from 1 to 12,

C_i = input concentration for ith batch,

V_i = measured volume of the precipitation feed makeup tank for the ith batch, and

H_{i-1} = holdup before ith batch is added.

Substituting this expression for T_{IN} into the equation for MB_1 gives

$$M_{B1} = \Sigma C_i V_i - \Sigma H_{i-1} + H_0 - H_{12}.$$

This particular transfer and inventory combination can be handled by considering $\Sigma C_i V_i$ as a correlated input transfer, ΣH_{i-1} as a correlated output transfer, and $(H_0 - H_{12})$ as a normal inventory. The $C_i V_i$ terms are not correlated with the H_i terms. However, the ΣH_{i-1} transfer is correlated with the inventory term $H_0 - H_{12}$. Because of the canceling effect of the $H_0 - H_{12}$ term, however, these inventory-transfer correlations will have a very small effect upon the materials balance variance in the steady-state case (see Appendix A) and are neglected in the propagated variance calculations.

The ADU cake is output at a frequency of 24 batches per day. Uranium concentration remains the same over each set of 24 batches, but each batch is weighed out separately. Therefore, the ADU cake transfer for 24 batches can be represented by

$$T_{OUT} = C\Sigma M_i \quad ,$$

where C is the concentration for all 24 batches, M_i is the wet cake mass for the ith batch, and Σ represents the summation

over i from 1 to 24. This type of transfer where the concentration remains constant for a set of batches is not directly available within the framework of the MEASIM code. However, the transfer of the type ΣC_iM_i (where the concentration C_i changes for each batch) is available. The ΣC_iM_i type transfer can be easily converted to the $C\Sigma M_i$ type transfer by setting the C_i uncorrelated error to zero and equating the uncorrelated and short-term correlated errors for C to the short-term and long-term correlated errors for C_i. For example, let

$$C = \mu(1 + \varepsilon + \eta) \quad ,$$

and

$$C_i = \mu(1 + \varepsilon_i + \eta_i + \theta_i) \quad ,$$

where

μ = true value,

ε_i = uncorrelated error,

η_i = short-term correlated error, and

θ_i = long-term correlated error.

Then ΣC_iM_i can be made equivalent to $C\Sigma M_i$ by setting

$\varepsilon_i = 0$,

$\eta_i = \varepsilon$, and

$\theta_i = \eta$,

and recalibrating the short-term correlated error (η_i) every 24 batches. In this case C_i will change every 24 batches just as C according to the uncorrelated error component of C. Also C_i and C will be governed by the same correlated errors.

In addition to the precipitation feed makeup unit, inventory measurements are also made for the precipitation, filter, and polish filter units. The third transfer for this UPAA is an output from the filtrate storage unit at a frequency of 10 batches per day. The amount of uranium in each batch is computed from the products of volume and concentration.

The process variables required for this example along with their nominal values and transfer batch frequencies are given in Table XII. Process variable 1 is labeled as the "Input Volume" in Table XII. This process variable is the volume measured in the precipitation feed makeup tank and includes the tank holdup volume as well as the input volume.

Materials balances are calculated once per day. Table XII shows that the transfers take place at a much higher frequency than this. The ADU cake has the highest frequency with 24 output batches/day. Because the MEASIM code requires that all the transfers have the same frequency, it is necessary to supplement the

TABLE XII

PROCESS VARIABLES FOR EXAMPLE 5

No.	Variable	Nominal Value	Frequency Batches/Day
1	Input vol. (L)	12.6	12
2	Input conc. (kg/L)	0.22	12
3	Holdup-precip. feed (kg)	0.022	-
4	Inventory-precipitator (kg)	0.1	-
5	Inventory-filter (kg)	0.9	-
6	Output mass-ADU cake (kg)	9.0	24
7	Output conc.-ADU cake (kg/L)	0.15148	24
8	Inventory-polish filter (kg)	0.22	-
9	Output vol.-filtrate storage (L)	35.0	10
10	Output conc.-filter storage (kg/L)	0.0008	10

other transfers with zeros to increase their frequency to 24 batches/day. This means that 12 zero transfers per day be included with the input transfer and 14 zero transfers per day with the filtrate storage output transfer.

The input file for this example is given in Fig. 18. A Monte Carlo simulation with 100 runs (NRUN = 100) and 20 materials balances (NBAL = 20) is selected at lines 11 and 12. The number of transfers per balance period (NTRPBL) is set at 24 in line 13. At line 27 for the inventory-transfer numbers, the input transfer corresponds to the first two entries, "6" and "-5," a result of the method used to model the input transfer as discussed above. Correspondingly, at lines 31 and 32 the precipitation feed holdup (process variable No. 3) is selected as the process variable for two different inventory transfers. The holdup is used as a transfer at line 31 and as an inventory at line 32. At line 56, the actual uncorrelated error for the ADU cake output concentration appears as the short-term correlated error with recalibration every 24 batches and the actual short-term correlated error appears as the long-term correlated error so modeling the transfer $C\Sigma M_i$ within the framework of the MEASIM code becomes possible.

The output file for this example is given in Fig. 19. For the "IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS," the number of process variables used (NPVCNT) is 11, even though there are only 10 process variables. The code has no way of knowing that one process variable, namely the precipitator feed holdup, is used twice. This discrepancy causes no computation problems, however, so long as the array sizes are set to handle 11 process variables. Similar reasoning applies to the number of random-number streams (NNSTRM). Counting the precipitator holdup twice causes the number of random-number seeds to be 20 instead of 18. Because there are only 17 nonzero measurement errors, it would be reasonable to expect only 17 random-number streams. However, the code always allocates a random-number stream for uncorrelated errors whether or not the error is zero. In this example, the uncorrelated error associated with the ADU cake concentration

```
/ ******* EXAMPLE 5 - CORRELATED INPUT TRANSFER AND INVENTORY
/
/ ******* PARAMETERS (HBALMX=105, NTRNMX=515, NPVMX=12, NPVIMX=10, NPVTMX=8,
/                    NCIMX=10, NCTMX=8, NMIMX=2, NMTMX=2, NSIMX=2, NSTMX=2,
/                    NCMX=2, MXPMX=2, NSTRMX=20, NCOLMX=2)
/
EXAMPLE 5 - CORRELATED INPUT TRANSFER AND INVENTORY
/
1/          IZE    - ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE)
0/          IRNSCH - RANDOM NUMBER SEEDS CHANGE FROM RUN TO RUN WHEN NONZERO
100/        NRUN   - NUMBER OF RUNS
20/         NBAL   - NUMBER OF MATERIALS BALANCES
24/         NTRPBL - NUMBER OF TRANSFERS PER BALANCE
0./         DT     - TIME CONSTANT (NOT USED IN THIS EXAMPLE)
0/          MASPRT - PRINTS MASSAGE DEBUGG OUTPUT WHEN NONZERO
0/          ITIPRP - PRINTS TRANSFER-INVENTORY SET NO. AND PROC. VAR. NO. IF=0
0/          IMESPR - PRINTS INPUT MEASUREMENT ERRORS WHEN EQUAL TO 1
0/          IPVPRT - PRINTS PROCESS VARIABLE FILE WHEN NON ZERO
2/          ICLAPS - REDUCES DIMENSIONS OF DECANAL INPUT FILE WHEN NONZERO
/
/ ********** INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA
/
10/         NPV    - NUMBER OF PROCESS VARIABLES
8/          NTRIN  - NUMBER OF TRANSFER AND INVENTORY SETS IN THE PROCESS
/
/   ITIN - INVENTORY-TRANSFER NUMBERS
6 -5 1 1 1 -6 1 -6/
/
/   IPVNO - PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER-INVENTORY
1 2/        INPUT TRANSFER (PRODUCT OF VOLUME AND  CONCENTRATION)
3/          INPUT TRANSFER (PRECIP. FEED HOLDUP CONTRIBUTION)
3/          HOLDUP - PRECIPITATOR FEED
4/          INVENTORY - PRECIPITATOR
5/          INVENTORY - FILTER
6 7/        OUTPUT - ADU CAKE
8/          INVENTORY - POLISH FILTER
9 10/       OUTPUT - FILTRATE STORAGE
/
0/          ISPNTI - SPECIFIC INVENTORY-TRANSFER SET NO. (0 GIVES ALL SETS)
/
/ *********** MEASUREMENT ERRORS ASSOCIATED WITH EACH PROCESS VARIABLE
/
1:  INPUT VOLUME
      .01 .005/
2:  INPUT CONCENTRATION
      .003 .001/
3:  HOLDUP - PRECIPITATOR FEED (INPUT TRANSFER CONTRIBUTION)
      .2 .1/
4:  INVENTORY - PRECIPITATOR
      .167/
5:  INVENTORY - FILTER
      .0833/
6:  OUTPUT MASS - ADU CAKE
      .002 .001 0 2/
7:  OUTPUT CONCENTRATION - ADU CAKE
      0 .067 .05 1 24/
8:  INVENTORY - POLISH FILTER
      .05/
9:  OUTPUT VOLUME
      .01 .005/
10: OUTPUT CONCENTRATION
      .1 .05/
```

Fig. 18.
Input data--Example 5.

```
INPUT DATA FILE - MSDT10
PV ARRAY FILE - PVAR10
OUTPUT DECANAL FILE - DECINN

TUE, OCT 20 1981     07:25:37

TITLE FROM PROCESS VARIABLE FILE

EXAMPLE 5 - CORRELATED INPUT TRANSFER AND INVENTORY

    ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) = 1
    FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) = 0
    NUMBER OF RUNS (NRUN) = 100
    NUMBER OF BALANCES (NBAL) = 20
    NUMBER OF TRANSFERS PER BALANCE (NTRPBL) = 24
    TIME INTERVAL (DT) =   0.900
    MASSAGE DEBUGG PRINT FLAG (MASPRT) = 0
    TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPRP) = 0
    PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) = 0
    PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) = 0
    ICLAPS (COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. 0) = 2


*****************************************************
INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)

    NUMBER OF PROCESS VARIABLES (NPV) = 10
    NUMBER OF TRANSFER-INVENTORIES (NTRIN) =  8
    ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)
      6 -5  1  1  1 -6  1 -6

    ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH TRANSFER OR INVENTORY (IPVNO)

         TRANSFER
         INVENTORY           PROCESS VARIABLE NUMBER
          NUMBER       (1)    (2)     (3)     (4)     (5)
            1           1      2       0       0       0
            2           3      0       0       0       0
            3           3      0       0       0       0
            4           4      0       0       0       0
            5           5      0       0       0       0
            6           6      7       0       0       0
            7           8      0       0       0       0
            8           9     10       0       0       0

    SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) =  0

*******************************************************
BEGIN READING IN PROCESS VARIABLE ARRAY

    NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY = 10
    IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES (1 FOR TRANSFER)
     1 1 1 0 0 1 1 0 1 1
    NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABLE
     515 515 515  25  25 515 515  25 515 515

READING OF PROCESS VARIABLE ARRAY COMPLETE
*******************************************************


MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE

    1:  INPUT VOLUME
        INITIAL VALUE =   12.599998
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.010000  0.005000  0.000000       1          10000        10000
    2:  INPUT CONCENTRATION
        INITIAL VALUE =     0.220000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.003000  0.021000  0.000000       1          10000        10000
    3:  HOLDUP - PRECIPITATOR FEED (INPUT TRANSFER CONTRIBUTION)
        INITIAL VALUE =     0.022000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.200000  0.100000  0.000000       1          10000        10000
    4:  INVENTORY - PRECIPITATOR
        INITIAL VALUE =     0.100000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.167000  0.000000  0.000000       1          10000        10000
    5:  INVENTORY - FILTER
        INITIAL VALUE =     0.900000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.083300  0.000000  0.000000       1          10000        10000
    6:  OUTPUT MASS - ADU CAKE
        INITIAL VALUE =     9.000000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.002000  0.001900  0.000000       2          10000        10000
    7:  OUTPUT CONCENTRATION - ADU CAKE
        INITIAL VALUE =     0.151480
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.000000  0.067000  0.050000       1             24        10000
    8:  INVENTORY - POLISH FILTER
        INITIAL VALUE =     0.220000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.050000  0.000000  0.000000       1          10000        10000
    9:  OUTPUT VOLUME
        INITIAL VALUE =    35.000000
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.010000  0.005000  0.000000       1          10000        10000
   10:  OUTPUT CONCENTRATION
        INITIAL VALUE =     0.000800
                  SIGMAE   SIGMAN(1)  SIGMAN(2)   MESTYP     INTCAL(1)       INTCAL(2)
                0.100000  0.050000  0.000000       1          10000        10000
```

Fig. 19.
Output file--Example 5.

****

IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS

```
NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT) = 11
NUMBER OF INVENTORY PROCESS VARIABLES (NPVI) =  4
NUMBER OF TRANSFER PROCESS VARIABLES (NPVT) =  7
NUMBER OF INVENTORY COMPONENTS (NCI) =  4
NUMBER OF TRANSFER COMPONENTS (NCT) =  4
NUMBER OF INVENTORY MEASUREMENTS (NMI) =  1
NUMBER OF TRANSFER MEASUREMENTS (NMT) =  2
NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI) =  1
NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST) =  2
NUMBER OF RANDOM NUMBER STREAMS (NNSTRM) = 20
NUMBER OF PULSE COLUMNS (NCOLUM) =  0
```

****

INITIAL RANDOM NUMBER SEEDS

```
1042839332   273785636  1973287924  1291185537  2063904014   164960264   491400187  1352537143   271703562  1141280589
 763526641   225695806  1653987289  1673092674  1567589403  1739823648  1986540383  2098769873  1796576548  1876940846
```

****

SUMMARY FOR ALL INVENTORIES AND TRANSFERS

****

I	XI	S2I	CVI	T	S2T	CVT	MAT. BAL.	S2XMB	CUSUM	S2CUSUM
0	1.2420E 00	6.0446E-03	4.8400E-06	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	2.9731E-04	7.5342E 00
2	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	5.9462E-04	2.0468E 01
3	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	8.9192E-04	3.8815E 01
4	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	1.1892E-03	6.2574E 01
5	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	1.4865E-03	9.1745E 01
6	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	1.7838E-03	1.2633E 02
7	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	2.0812E-03	1.6632E 02
8	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	2.3785E-03	2.1173E 02
9	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	2.6758E-03	2.6255E 02
10	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	2.9731E-03	3.1878E 02
11	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	3.2704E-03	3.8042E 02
12	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	3.5677E-03	4.4748E 02
13	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	3.8650E-03	5.1995E 02
14	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	4.1623E-03	5.9782E 02
15	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	4.4596E-03	6.8112E 02
16	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	4.7569E-03	7.6982E 02
17	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	5.0542E-03	8.6393E 02
18	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	5.3515E-03	9.6346E 02
19	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	5.6489E-03	1.0684E 03
20	1.2420E 00	6.0446E-03	4.8400E-06	2.9731E-04	7.5221E 00	2.7060E 00	2.9731E-04	7.5342E 00	5.9462E-03	1.1787E 03

****

RESULTS OF MONTE CARLO SIMULATION WITH   100 SAMPLES

****

CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE

```
UPPER LIMIT =   1.29223
LOWER LIMIT =  0.736473
```

BALANCE NUMBER	CUSUM SAMPLE AVERAGE	CUSUM PROPAGATED VARIANCE	CUSUM SAMPLE VARIANCE	RATIO SAMPLE/ PROPAGATED	
1	-3.161461D-01	7.53416	9.38718	1.24595	OK
2	-5.869121D-01	20.4684	27.0089	1.31954	***** RATIO OUTSIDE INTERVAL
3	-7.805135D-01	38.8149	42.3207	1.09032	OK
4	-1.131021D 00	62.5737	67.4099	1.07729	OK
5	-9.809849D-01	91.7445	101.419	1.10545	OK
6	-1.156011D 00	126.327	142.970	1.13174	OK
7	-1.353591D 00	166.322	201.182	1.20959	OK
8	-1.635189D 00	211.729	251.653	1.18856	OK
9	-2.026552D 00	262.548	315.435	1.20144	OK
10	-2.004099D 00	318.779	393.634	1.23482	OK
11	-2.319147D 00	380.423	471.675	1.23987	OK
12	-2.408884D 00	447.478	557.128	1.24504	OK
13	-2.566654D 00	519.945	654.563	1.25891	OK
14	-2.685358D 00	597.824	751.526	1.25710	OK
15	-2.803760D 00	681.116	844.364	1.23968	OK
16	-2.783470D 00	769.818	953.715	1.23888	OK
17	-3.195573D 00	863.933	1080.60	1.25079	OK
18	-3.037237D 00	963.461	1189.90	1.23503	OK
19	-3.464316D 00	1068.40	1303.69	1.22022	OK
20	-3.316118D 00	1178.75	1444.85	1.22575	OK

Fig. 19. (cont)

is "artificially" set to zero to allow for modeling of the output transfer as previously discussed. The results of the Monte Carlo simulation show good agreement between the sampled and propagated CUSUM variances with only one ratio outside the 95% confidence interval.

VIII. COMPUTER REQUIREMENTS

The MEASIM code has been checked out and run on a PRIME 750 computer. Computer core memory requirements are relatively high because of the large arrays required for process variables, transfers and inventory components, and variances and covariances. The code itself requires 22 148 16-bit words of computer memory to load. An additional 67 819 words are required for common-block storage giving a total of 89 967 words for the complete program with the capability for modeling a maximum of 105 materials balances and 515 transfers.

Computation times are very much problem dependent. A summary of the computer Central Processor Unit (CPU) time for the 5 example problems in this manual is given in Table XIII.

TABLE XIII

CPU TIME REQUIREMENTS FOR THE EXAMPLE PROBLEMS

Example	CPU[a] Time (s)
1	3.5
2	210.2
3	939.1
4	34.0
5	519.1

[a]PRIME 750 Central Processor Unit.

Example problem 1 consists of one zero-error run with 50 balances and 50 transfers. The larger computation times associated with examples 2-5 are due to additional transfer components and to the Monte Carlo runs. Additional transfers have significant effects upon computation times because the code must check for the numerous correlation possibilities for the transfers. Although example problem 3 has only 10 balances, the computation time is relatively large because there are ten transfers per materials balance, and Monte Carlo runs are made for the entire UPAA and for each individual transfer and inventory element. In addition, both input and output transfers are computed from the product of flow rate and concentration. Because each of these transfers requires 4 transfer components, the large number of transfer correlations causes the computation time to be increased. Example 5 has only 20 materials balances but 24 batch transfers per balance period.

IX. OPERATIONAL PROCEDURES

This section provides guidelines for computer implementation of the code and a suggested procedure for measurement modeling a given process.

The entire source code resides on one file and is written with very standard FORTRAN IV for a PRIME 750 computer. Possibly the only nonstandard feature of the code is the use of the FORTRAN PARAMETER statement to dimension the arrays. The PARAMETER statement makes it possible to change the problem dimensions very quickly and easily. This capability is particularly advantageous for the inexperienced user. For those machines that do not support the PARAMETER statement, it would be necessary for the user to replace the parameters with integers in the COMMON and DIMENSION statements. In addition, subroutines WRTB, WRTC, WRTR, WRTS, WRT3, and WRT5 called from subroutines MASAGE and SETMAS have parameters in their calling arguments. Also, some DO-loop

indices at the beginning of subroutine SETMAS are defined with parameters making it necessary to convert these to integers.

Subroutine DRAND computes uniform random numbers over the range 0-1 and is highly computer dependent. For best results, a user should replace the random-number generator in DRAND with one compatible with his or her computer.

In applying the code to a given UPAA, the first step is to define the process variables required to compute a materials balance. These process variables could be generated by either a process model code[1] or, in simple steady-state cases with no process variations, by a small code specifically designed to write arrays of constant values. For the examples in this manual, the process variables were generated from a small, special-purpose code given in Appendix C. The process variables are input to the MEASIM code on a separate file.

With process variables defined, the next step is to develop the input data file. It may prove to be easier to start at the end of the file with the measurement error data. Here each process variable requires the σ's for uncorrelated and correlated errors as well as the measurement type (see MESTYP in Table VI) and recalibration frequencies for the correlated errors. The order of appearance for this measurement data must be consistent with the numbering of process variables as defined by the process variable file.

The next recommended step is to establish the input data defining the process perhaps by first defining each inventory and transfer set in the UPAA through the ITIN array (see Table VI). The order of occurrence in the ITIN array defines the numbering system for the inventory and transfer sets. With ITIN established, the remaining inputs required to define the UPAA should be relatively easy to determine. The number of process variables is NPV, whereas the number of transfer and inventory sets is NTRIN; that is, the number of elements in ITIN. The process variables associated with each transfer and inventory set are input

to the IPVNO array by means of the user-defined numbering system. If the user desires to calculate the entire UPAA, ISPNTI should be set to zero. For isolated calculations of one transfer or inventory set, ISPNTI should be set to the number of that particular set.

In the first portion of the input data, the number of runs (NRUN) should be set to 1 for a normal measurement calculation and to a value greater than one for a Monte Carlo simulation. NBAL specifies the number of balances and NTRPBL the number of transfer measurements per balance period. All transfers in a given UPAA must have the same frequency. The input DT defines the time interval between flow-rate calculations and is of importance only for transfers computed from the product of flow rate and concentration.

The third input file to the MEASIM code contains the random-number seeds. Each seed is a positive ten-digit integer less than or equal to 2 147 483 648. The number of seeds must be equal to or greater than the number of nonzero measurement errors with the seeds entered one per line.

With the three input files constructed as above, the next step is to compile the MEASIM code. Before compiling, however, it is advisable to check the parameter values in the FORTRAN PARAMETER statement to make certain that the dimensions of the arrays are adequate to handle the problem under consideration. The parameters are defined in Table I.

After checking the parameters, the code can be compiled, link loaded, and then run. On the PRIME 750 computer, these operations can be performed with the following statements.

```
(     1) FTN MEASIM -64V -L
(     2) SEG
(     3) VLOAD #MEASIM
(     4) LO B_MEASIM
(     5) LIB VAPPLB
(     6) LI
(     7) MA MAP 0
(     8) QU
(     9) SEG #MEASIM
```

(10) MESDT1
(11) PVARA1
(12) DECINN
(13) SPOOL MESOUT
(14) CO TTY

Line 1 compiles the code, lines 2-8 load the code, and lines 9-13 run the code and transfer the output file to the high-speed printer. A complete FORTRAN listing of MEASIM appears in Appendix D.

X. ACKNOWLEDGMENTS

The author wishes to thank all of the Safeguards Systems personnel who contributed to this user's manual. H. A. Dayem developed an early version of the code. J. T. Markin and J. P. Shipley developed subroutine MASAGE, which was very important in attaining the high level of process independence for the MEASIM code. A. S. Goldman provided some valuable insight for the chi-squared confidence interval tests. M. J. Roybal and M. L. Bonner prepared the figures, and K. C. Eccleston and S. L. Hurdle typed the manuscript.

APPENDIX A

INVENTORY CORRELATED ERRORS AND THE CUSUM VARIANCE

As indicated in Sec. III.E, calculation of the CUSUM variance is a very important tool for program debugging. The definition for the variance calculation is given in Eq. (6). For large UPAAs with many correlated errors, this calculation can become very complex. To simplify these calculations, the MEASIM code neglects correlation between two different inventories and between most inventories and transfers. The purpose of this appendix is to provide some justification for these simplifications.

The UPAA shown in Fig. A-1 with two transfers and two inventories will be adequate for this development. From Eq. (2), it follows that the CUSUM for this system over the time interval 0-t with N input and output transfers taking place during this time will be

$$\mathrm{CUSUM} = I_1(o) + I_2(o) - I_1(t) - I_2(t) + \Sigma(T_{1i} - T_{2i}) \quad , \quad \text{(A-1)}$$

where Σ indicates a summation from i = 1 to N.

For this development it will be assumed that

(1) measurement errors behave according to a multiplicative error model,

(2) uncorrelated errors are zero,

(3) only one correlated error is present in each measurement, and

(4) each inventory and transfer consists of one component with one measurement per component.

The above assumptions do not limit the conclusions but only help to simplify the resulting equations. Under these assumptions with I_m and T_M measured inventories and transfers, respectively, and with I and T the corresponding true values, it follows from Eq. (3) that

$$I_m = I(1 + \eta_I) \quad , \quad \text{and} \tag{A-2}$$

$$T_m = T(1 + \eta_T) \quad . \tag{A-3}$$

When the above expressions for inventory and transfer are substituted into Eq. (A-1) for the CUSUM, and the resulting CUSUM is substituted for X in Eq. (6), the true values of the inventories and transfers combine to form the true CUSUM value. Because all the measurement errors are zero mean, the mean value μ_x in the CUSUM variance equation is cancelled by all the true inventory and transfer values and leads to the following equation for the CUSUM variance, σ_c^2.

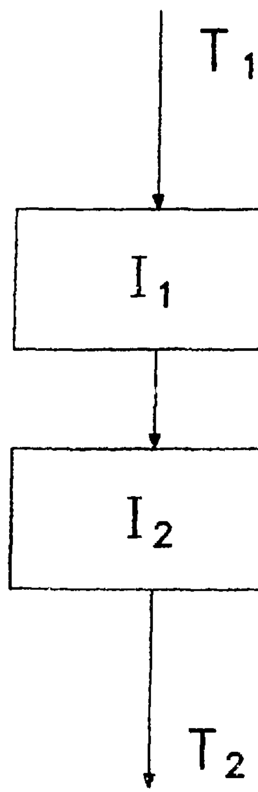

Fig. A-1.
Process block diagram--Example UPAA.

$$\sigma_c^2 = E(\{[I_1(o) - I_1(t)]\eta_{I1} + [I_2(o) - I_2(t)]\eta_{I2}$$

$$+ \Sigma(T_{1i}\eta_{T1} - T_{2i}\eta_{T2})\}^2) \quad . \tag{A-4}$$

In Eq. (A-4), the maximum number of terms is present when all measurements are made with the same instrument. Hence, all the correlated errors are equal; that is,

$$\eta_{I1} = \eta_{I2} = \eta_{T1} = \eta_{T2} = \eta \quad , \quad \text{and}$$

$$E\{\eta\eta\} = \sigma_\eta^2 \quad .$$

Eq. (A-4) yields

$$\sigma_c^2 = [K_1 + K_2 + K_3 + K_4 + K_5]\sigma_\eta^2 \quad , \tag{A-5}$$

where

$K_1 = [I_1(o) - I_1(t)]^2 + [I_2(o) - I_2(t)]^2$;

$K_2 = 2[I_1(o) - I_1(t)][I_2(o) - I_2(t)]$;

$K_3 = 2[I_1(o) - I_1(t)](\Sigma T_{1i} - \Sigma T_{2i})$;

$K_4 = 2[I_2(o) - I_2(t)](\Sigma T_{1i} - \Sigma T_{2i})$; and

K_5 = function of transfers only.

If the process is operating near a steady-state condition, then the initial and final inventories will be almost equal. In addition, the summation of the input and output transfers will be almost equal, that is,

$I_1(o) = I_1(t)$,

$I_2(o) = I_2(t)$, and

$\Sigma T_{1i} \approx \Sigma T_{2i}$.

Under these conditions, the constants K_1, K_2, K_3, and K_4 will be close to zero; hence, correlated inventory errors will have very little effect upon the CUSUM variance. This important conclusion is valid for the simple UPAA shown in Fig. A-1. Similar conclusions could be derived for more complicated UPAAs under steady-state operating conditions.

Two zero-mean random variables X and Y are said to be correlated if[3]

$$E\{XY\} \neq 0.$$

Thus, if inventories I_1 and I_2 have the same correlated error η then

$$E\{I_1\eta I_2\eta\} = I_1 I_2 \sigma_\eta^2 \quad ,$$

and the two inventories are correlated. Similarly, if an inventory and transfer have the same correlated error, then they are correlated.

In deriving Eq. (A-5), all the inventories and transfers were assumed to have the same correlated error. Thus, all the inventories and transfers are correlated. Among the constants K_i in Eq. (A-5), K_2 is caused by correlations between different inventories, whereas K_3 and K_4 result from inventory and transfer correlations. Inventory variances caused by correlated errors are taken into account by K_1.

The MEASIM code includes effects of correlated errors in the inventory variance calculations; that is, K_1 is not assumed to be zero. However, if initial and final inventories are almost equal, then K_1 will be small. Hence, the inventory variance from correlated errors will be small and will have a relatively small effect upon the CUSUM variance.

MEASIM's neglect of correlation between different inventories and most correlations between inventories and transfers implies that K_2, K_3, and K_4 are zero; K_2 will be small if either of the inventories I_1 or I_2 have initial and final values that are almost equal. Constant K_3 will be small if the change in inventory 1 is small or if the net input transfer, ΣT_{1i}, is approximately equal to the net output transfers, ΣT_{2i}. Similar reasoning applies to I_2 and K_4.

APPENDIX B

CONFIDENCE INTERVALS FOR MONTE CARLO SIMULATIONS

As indicated in Sec. V and VI, the Monte Carlo option is very valuable for verifying the correctness of the simulation. The MEASIM code, when running in the Monte Carlo mode, makes one zero-error run for computing the propagated or analytic CUSUM standard deviation σ and then a series of runs with measurement errors to calculate the sampled CUSUM σ. Both propagated and sampled σ's are computed for each materials balance. If sampled CUSUM σ's are within some confidence interval of the propagated σ's, then some assurance is provided that the propagated CUSUM σ is being calculated correctly and that measurement errors are being correctly modeled. The purpose of this appendix is to derive the confidence intervals that form the basis for comparing the propagated and sample standard deviations. Most of the theory for this development is taken from Ref. 4.

Let X_1, X_2, ..., X_n denote the values of the CUSUM at some fixed materials balance for n different runs. The sample mean X is defined by

$$\bar{X} = \frac{1}{n} \Sigma X_i \quad , \tag{B-1}$$

where Σ indicates summation over i from 1 to n, and the sample variance S^2 (unbiased estimate) is

$$S^2 = \frac{1}{n-1} \Sigma (X_i - \bar{X})^2 \quad . \tag{B-2}$$

Because the CUSUM or X_i consists of the sum of random variables, it is reasonable to conclude from the Central Limit Theorem[4] that X_i is a normally distributed random variable.

For normally distributed X_i and with σ^2, the true variance of the entire X_i population, it follows from Ref. 4 that the random variable $(n - 1)S^2/\sigma^2$ is chi-squared distributed with n - 1 degrees of freedom. The true variance σ^2 is equal to the propa-propagated or analytic variance calculated by the MEASIM code.

For a 95% confidence interval, it follows from Ref. 4 that

$$\chi^2_{2.5} \leq \frac{(n - 1)S^2}{\sigma^2} \leq \chi^2_{97.5} \quad , \tag{B-3}$$

where χ^2_p is the pth percentile value for the chi-squared distribution. For example, $\chi^2_{97.5}$ defines a point along the horizontal axis such that 97.5% of the area under the chi-squared density curve lies to the left of that point.

The percentile values for the chi-squared distribution are readily available from tables. However, when the number of degrees of freedom df is >30, the chi-squared percentile values can be calculated to a reasonable degree of accuracy by a relatively simple analytic approach. For df > 30, the function $(2\chi^2_p)^{1/2} - (2df - 1)^{1/2}$ is almost normally distributed with mean zero and standard deviation one.[4] If z_p is the pth percentile of the standardized normal distribution, it follows that

$$z_p = (2\chi^2_p)^{1/2} - (2df - 1)^{1/2} \quad .$$

Solving for χ^2_p gives

$$\chi^2_p = 1/2(z_p + \sqrt{2df - 1})^2 \quad . \tag{B-5}$$

For a 95% confidence interval, 2.5% of the area under the density function curve must lie in each of the two tails requiring percentiles of 2.5 and 97.5. For the standard normal distribution

$$z_{2.5} = -1.96 \quad , \quad \text{and}$$

$$z_{97.5} = 1.96 \quad .$$

Substituting these percentiles into Eq. (B-6) results in the following chi-squared percentiles for a 95% confidence interval

$$\chi^2_{2.5} = \frac{1}{2}(-1.96 + \sqrt{2df - 1})^2 \quad , \quad \text{and} \tag{B-6}$$

$$\chi^2_{97.5} = \frac{1}{2}(1.96 + \sqrt{2df - 1})^2 \quad . \tag{B-7}$$

The above expressions are approximations to the chi-squared percentiles valid for the number of degrees of freedom df at least 30 or greater. A comparison of these approximate chi-squared percentiles with the exact tabular values[5] is given in Table B-I for different degrees of freedom, df. Inspection of Table B-I shows that Eqs. (B-6) and (B-7) are good approximations to the exact tabular values for degrees of freedom in the 30-1000 range.

Substituting (n - 1) for df in Eqs. (B-6) and (B-7), substituting the resulting χ^2 expressions into Eq. (B-3), and dividing by (n - 1) gives

$$\frac{[-1.96 + \sqrt{2(n-1) - 1}]^2}{2(n-1)} \leq \frac{S^2}{\sigma^2} \leq \frac{[1.96 + \sqrt{2(n-1) - 1}]^2}{2(n-1)} \quad . \tag{B-8}$$

TABLE B-I

COMPARISON OF APPROXIMATE CHI-SQUARED PERCENTILES WITH EXACT TABULAR VALUES

Degrees of Freedom (df)	$\chi^2_{2.5}$ Table	$\chi^2_{2.5}$ Eq. (B-6)	$\chi^2_{97.5}$ Table	$\chi^2_{97.5}$ Eq. (B-7)
30	16.8	16.4	47.1	46.5
100	74.2	73.8	130.0	129.1
500	440.0	439.5	565.0	563.4
1000	914.0	913.8	1090.0	1089.1

Inequality (B-8) forms the basis for computing 95% confidence limits on the S^2/σ^2 ratio. Recall that S^2 is the unbiased sample CUSUM variance computed from n samples, while σ^2 is the corresponding propagated variance.

The MEASIM code performs measurement modeling for a given UPAA for a number of materials balances as specified by the NBAL input integer. A CUSUM is calculated at the end of each materials balance along with the S^2/σ^2 ratio. Thus, NBAL materials balances for a given UPAA will yield an equal number of S^2/σ^2 ratios. Suppose, for example, that there are 100 balances (NBAL = 100). It might be easy to conclude that with a 95% confidence interval that about 5% of the S^2/σ^2 ratios for a given Monte Carlo simulation of n runs will lie outside the boundary. However, this is an incorrect interpretation of the confidence intervals in this particular application. The difficulty here is that the sample variances S^2 are strongly correlated from balance to balance. For example, the CUSUM at the kth materials balance is the sum of the first k materials balances, while the CUSUM at the (k + 1)th materials balance is equal to the CUSUM at the kth balance plus the (k + 1)th materials balance. Therefore, the sample variance S^2 cannot change significantly between the kth and (k + 1)th balance and leads to situations in a given Monte Carlo simulation where all the S^2/σ^2 ratios

are clustered in one region, either inside or outside of the confidence interval, for all NBAL CUSUMs. To correctly interpret the confidence intervals for this case, attention must be focused on a CUSUM calculated from a given number of materials balances, for example, k balances. Let the sample variance be S_k^2 for this CUSUM. A 95% confidence interval indicates that if the Monte Carlo simulation is repeated a large number of times, the ratio S_k^2/σ^2 will fall within that interval about 95% of the time. In other words, the confidence intervals must be interpreted at a fixed number of balances over a sufficiently large number of Monte Carlo runs and not over the balances within any one Monte Carlo simulation.

Suppose, for example, with a 95% confidence interval that the S^2/σ^2 ratio lies outside the confidence interval for all the balance times. Because of the strong correlation between the ratios, additional sets of Monte Carlo runs should be made before any conclusions are drawn concerning the correctness of the measurement modeling and variance propagation.

A relatively lengthy computer run was conducted to lend support to the above discussion. The process of Example 1 was used for this case. A total of 1000 sets of Monte Carlo simulations were performed, with each Monte Carlo set consisting of 100 calculations of the CUSUM at each of the 50 materials balances. For each Monte Carlo set of 100 runs it was determined at each materials balance whether or not the ratio S^2/σ^2 was inside or outside the confidence interval. This Monte Carlo simulation was then repeated 1000 times. At each materials balance, a count was made of the number of times the ratio S^2/σ^2 was outside the confidence interval.

The results from this simulation are given in Fig. B-1. For a 95% confidence interval at each materials balance, the percentage of S^2/σ^2 ratios outside the confidence interval theoretically should be 5%. The results shown in Fig. B-1 appear to support the theory for this case.

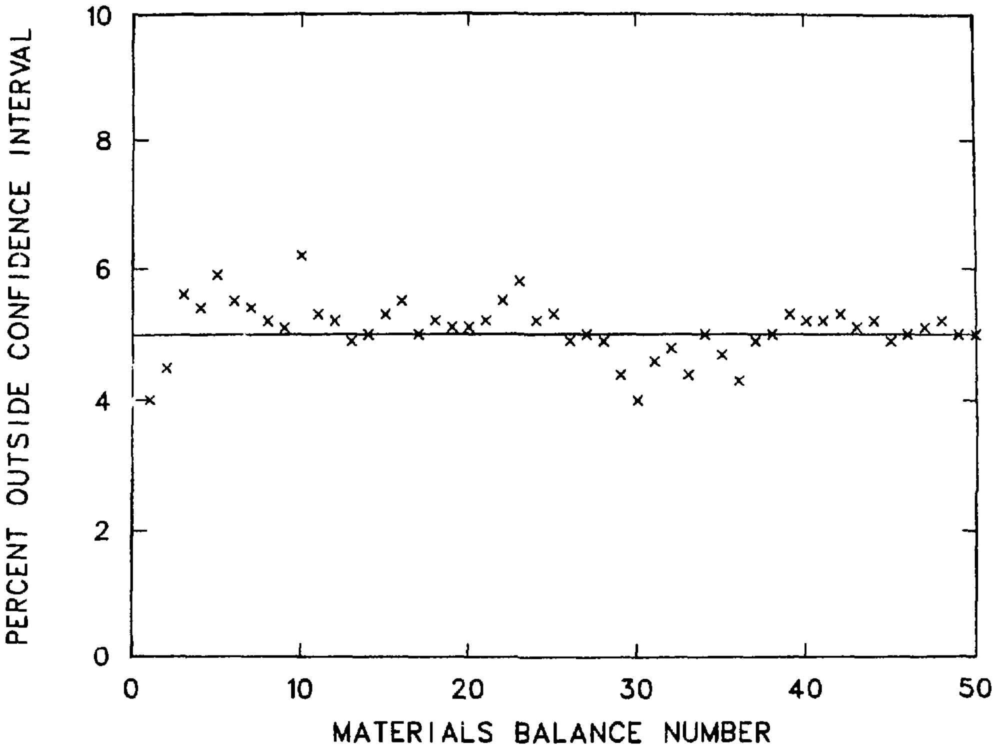

Fig. B-1.
Per cent of the s^2/σ^2 ratios lying outside the 95% confidence interval.

APPENDIX C

PROCESS VARIABLE FILES

Normally, the process variable files are generated by the process model code. However, for simple models with zero-process variations, the process variables can be generated directly with a small, special-purpose computer code. The process variables for the five examples in this user's manual were generated this way to make it relatively easy for a potential user to generate the process variable file for code checkout.

The code required to generate process variables for the five examples in this user's manual is called PVGEN. A listing of PVGEN is given in Fig. C-1. PVGEN requires an input data file consisting of five input variables summarized in Table B-1 in the order that they must appear on the input file.

Of these five input variables, only the transfer indicator, IPVTRN, requires further explanation. Each of the NPV entries in the IPVTRN array corresponds to a single process variable. IPVTRN(1) corresponds to the first process variable, IPVTRN(2) to the second process variable, etc. A zero IPVTRN indicates an inventory process variable, whereas a nonzero value indicates a transfer process variable. For positive values of IPVTRN, for example N, only every Nth value of the corresponding transfer process variable will be nonzero. As an example, if IPVTRN(I) is equal to 10, then for the Ith process variable, every 10th value will be nonzero with all the other values zero. Negative entries for IPVTRN still indicate a transfer but with a different sequencing of zero and nonzero values. The negative IPVTRN with the largest absolute value establishes the number of transfers over a given repeating cycle and indicates that all the process variable values are nonzero in that cycle. The remaining negative IPVTRN entries with smaller absolute values specify the number of nonzero values for the corresponding process variable in that cycle. For example, if

IPVTRN(2) = -12,

IPVTRN(5) = -24, and

IPVTRN(7) = -16,

then the number of transfer process variable values per cycle is 24, process variable 5 has all nonzero values, process variable 2 has 12 nonzero values, and process variable 7 has 16 nonzero values.

The PVGEN code input data files for Example problems 1-5 are given in Figs. C-2 through C-6, respectively. Each data file contains the five input variables in the order shown in Table C-I.

Similar to the MEASIM code, the array sizes are set in the PVGEN code by a FORTRAN PARAMETER statement. The current values of the parameters are sufficient to handle the 5 examples in this manual.

Upon executing the code, the user is required to enter the file names for the input data file and the output process variable file. This output file then serves as the input process variable file to the MEASIM code.

TABLE C-I

INPUTS TO PROCESS VARIABLE CODE PVGEN

NPV/	Number of different process variables (PVs)
NINV/	Number of values for each inventory PV
NTRN/	Number of values for each transfer PV
IPVTRN(1), IPVTRN(2), ..., IPVTRN(NPV)	Transfer indicator
PV(1), PV(2), ..., PV(NPV)/	Process variable values

```
C **************************** PROGRAM PVGEN ******************************
C
C ********* COMPUTES PROCESS VARIABLE ARRAYS FOR INPUT TO MEASIM CODE
C
C **** INPUTS IN ORDER OF OCCURRENCE (FREE FORMAT)
C              NPV     - NUMBER OF PROCESS VARIABLES
C              NINV    - NUMBER OF INVENTORIES FOR EACH INVENTORY P.V.
C              NTRN    - NUMBER OF TRANSFERS FOR EACH TRNASFER P.V.
C              (IPVTRN(I),I=1,NPV)  - TRANSFER INDICATOR ARRAY
C                                    (1 INDICATES A TRANSFER)
C              (C(I),I=1,NPV) - PROCESS VARIABLE VALUES
C
      PARAMETER (NPVMX=12,NPVIMX=10,NPVTMX=8,NINVMX=105,NTRNMX=515)
C
C **** PARAMETER DEFINITIONS:
C          NPVMX  - MAXIMUM NUMBER OF PROCESS VARIABLES
C          NPVIMX - MAXIMUM NUMBER OF INVENTORY PROCESS VARIABLES
C          NPVTMX - MAXIMUM NUMBER OF TRANSFER PROCESS VARIABLES
C          NINVMX - MAXIMUM NUMBER OF INVENTORIES FOR EACH INVENTORY P.V.
C          NTRNMX - MAXIMUM NUMBER OF TRANSFERS FOR EACH TRANSFER P.V.
C
      DIMENSION C(NPVMX), PVI(NINVMX,NPVIMX), PVT(NTRNMX,NPVTMX),
     +  NPVCT(NPVMX), IPVTRN(NPVMX), ITITLE(40), NMDTRN(NPVTMX),
     +  IFT(NPVTMX)
C
      INTEGER*2 PVARA(4), PVDAT(4)
      WRITE(1,10)
   10 FORMAT(' ENTER NAMES FOR'/10X'1. INPUT DATA FILE'
     1 /10X'2. OUTPUT PROCESS VARIABLE ARRAY FILE'
     2 /' (ONE NAME PER FILE WITH A MAXIMUM OF 8 CHARACTERS')
      READ(1,20) (PVDAT(I),I=1,4)
      READ(1,20) (PVARA(I),I=1,4)
   20 FORMAT(4A2)
C
C *************** OPEN FILES
C
      CALL SRCH$$(1,PVDAT,8,1,0,0)
      CALL SRCH$$(5,PVARA,8,2,0,0)
      CALL SRCH$$(2,PVARA,8,2,0,0)
C
C *************** READ INPUT DATA
C
      READ(5,*)NPV
      READ(5,*)NINV
      READ(5,*)NTRN
      IF(NPV.LE.NPVMX)GO TO 40
      IDUM=NPVMX
   30 FORMAT(' INPUT NPV IS GREATER THAN NPVMX ='I4)
      GO TO 160
   40 READ(5,*)(IPVTRN(I),I=1,NPV)
      READ(5,*)(C(I),I=1,NPV)
C
C *************** ECHO CHECK OF INPUT DATA
C
      WRITE(1,50) NPV, NINV, NTRN
   50 FORMAT(//' NUMBER OF PROCESS VARIABLES (NPV) ='I3
     1 /' NUMBER OF INVENTORIES (NINV) ='I3
     2 /' NUMBER OF TRANSFERS (NTRN) ='I3)
      WRITE(1,60) (IPVTRN(I),I=1,NPV)
   60 FORMAT(//' TRANSFER INDICATOR ARRAY (1 INDICATES TRANSFER)
     1 / (20I3))
      WRITE(1,70) (C(I),I=1,NPV)
   70 FORMAT(//' PROCESS VARIABLE VALUES'/(6G12.5))
C
C *************** CHECK FOR ARRAY OVERFLOW POSSIBILITIES
```

Fig. C 1.
FORTRAN listing of PVGEN.

```
C
      IF(NINV.LE.NINVMX)GO TO 90
      IDUM=NINVMX
      WRITE(1,80)IDUM
   80 FORMAT(//' NUMBER OF INVENTORIES (NINV) IS LARGER THAN NINVMX ='
     1 I4)
      GO TO 160
   90 IF(NTRN.LE.NTRNMX)GO TO 110
      IDUM=NTRNMX
      WRITE(1,100)IDUM
  100 FORMAT(//' NUMBER OF TRANSFERS (NTRN) IS LARGER THAN NTRNMX ='I4)
      GO TO 160
  110 CONTINUE
C
C *************** COUNT UP NUMBER OF TRANSFER AND INVENTORY P.V.'S
C
      NTRNPV=0
      DO 120 I=1,NPV
      IF(IPVTRN(I).NE.0)NTRNPV=NTRNPV+1
  120 CONTINUE
      NINVPV=NPV-NTRNPV
      IF(NTRNPV.LE.NPVTMX)GO TO 140
      IDUM=NPVTMX
      WRITE(1,130)IDUM
  130 FORMAT(//' NUMBER OF TRANSFER P.V.S IS LARGER THAN NPVTMX ='I4)
      GO TO 160
  140 IF(NINVPV.LE.NPVIMX)GO TO 180
      IDUM=NPVIMX
      WRITE(1,150)IDUM
  150 FORMAT(//' NUMBER OF INVENTORY P.V.S IS LARGER THAN NPVIMX ='I4)
  160 WRITE(1,170)
  170 FORMAT(//'*************** RUN TERMINATED **************')
      GO TO 350
C
  180 CONTINUE
      WRITE(1,190)
  190 FORMAT('ENTER TITLE')
      READ(1,200)(ITITLE(I),I=1,40)
  200 FORMAT(40A2)
      WRITE(6,200) (ITITLE(I),I=1,40)
C
C ************ CHECK FOR IPVTRN < 0
C
      IPTNEG=0
      DO 210 I=1,NPV
      IF(IPVTRN(I).LT.0)IPTNEG=I
  210 CONTINUE
      IF(IPTNEG.EQ.0)GO TO 240
      DO 220 I=1,NPV
      IF(IPVTRN(I).EQ.0)GO TO 220
      IPVT=IPVT+1
      IFT(IPVT)=-IPVTRN(I)
      IPVTRN(I)=1
  220 CONTINUE
C
C **************** FIND LARGEST INTEGER IN IFT ARRAY
C
      MAX=-2000
      DO 230 I=1,NTRNPV
      IF(IFT(I).GT.MAX) MAX=IFT(I)
  230 CONTINUE
  240 CONTINUE
      NDUM=NPV
      IPVT=0
      DO 250 I=1,NPV
```

Fig. C-1. (cont)

```
      IF(IPVTRN(I).EQ.0)GO TO 250
      IPVT=IPVT+1
      NMDTRN(IPVT)=0
      IF(IPVTRN(I).EQ.1)GO TO 250
      NMDTRN(IPVT)=IPVTRN(I)
      IPVTRN(I)=1
  250 CONTINUE
      WRITE(6) NDUM
      WRITE(6) IPVTRN
      IPVI=0
      IPVT=0
      DO 320 I=1,NPV
      IF(IPVTRN(I).EQ.1)GO TO 270
      NPVCT(I)=NINV
      IPVI=IPVI+1
      DO 260 J=1,NINV
      PVI(J,IPVI)=C(I)
  260 CONTINUE
      GO TO 320
  270 NPVCT(I)=NTRN
      IPVT=IPVT+I
      IF(IPTNEG.EQ.0)GO TO 300
      ICNT=0
  280 DO 290 J=1,MAX
      ICNT=ICNT+1
      IF(ICNT.GT.NTRN)GO TO 320
      PVT(ICNT,IPVT)=C(I)
      IF(J.GT.IFT(IPVT))PVT(ICNT,IPVT)=0.
  290 CONTINUE
      GO TO 280
  300 CONTINUE
      DO 310 J=1,NTRN
      PVT(J,IPVT)=C(I)
      IF(NMDTRN(IPVT).EQ.0)GO TO 310
      IF(MOD(J,NMDTRN(IPVT)).NE.0)PVT(J,IPVT)=0
  310 CONTINUE
  320 CONTINUE
      WRITE(6) NPVCT
      IPVI=0
      IPVT=0
      DO 340 I=1,NPV
      NPVC=NPVCT(I)
      IF(IPVTRN(I).EQ.1)GO TO 330
      IPVI=IPVI+1
      WRITE(6) (PVI(J,IPVI),J=1,NPVC)
      GO TO 340
  330 CONTINUE
      IPVT=IPVT+1
      WRITE(6) (PVT(J,IPVT),J=1,NPVC)
  340 CONTINUE
  350 CONTINUE
      CALL SRCH$$(4,PVDAT,8,1,0,0)
      CALL SRCH$$(4,PVARA,8,2,0,0)
      CALL EXIT
      END
```

Fig. C-1. (cont)

```
(1)    8/            NPV - NUMBER OF PROCESS VARIABLES
(2)    53/           NINV - NUMBER OF VALUES PER INVENTORY PROCESS VARIABLE
(3)    55/           NTRN - NUMBER OF VALUES PER TRANSFER PROCESS VARIABLE
(4)    1 1 Ø Ø Ø Ø 1 1/  IPVTRN - TRANSFER INDICATOR ARRAY
(5)    1ØØ .Ø5 1ØØ .Ø5 1ØØ .Ø5 .Ø1 4.99/ PROCESS VARIABLE VALUES
```

Fig. C-2.
Input data to PVGEN--Example 1.

```
(1)    3/            NPV - NUMBER OF PROCESS VARIABLES
(2)    55/           NINV - NUMBER OF VALUES PER INVENTORY PROCESS VARIABLE
(3)    515/          NTRN - NUMBER OF VALUES PER TRANSFER PROCESS VARIABLE
(4)    1 Ø 1Ø/       IPVTRN - TRANSFER INDICATOR ARRAY
(5)    1 1ØØ 1Ø/     PROCESS VARIABLE VALUES
```

Fig. C-3.
Input data to PVGEN--Example 2.

```
(1)    11/           NPV - NUMBER OF PROCESS VARIABLES
(2)    12/           NINV - NUMBER OF VALUES PER INVENTORY PROCESS VARIABLE
(3)    115/          NTRN - NUMBER OF VALUES PER TRANSFER PROCESS VARIABLE
(4)    1 1 Ø Ø Ø Ø Ø Ø Ø 1 1/  IPVTRN - TRANSFER INDICATOR ARRAY
(5)    1Ø6 .Ø542 .ØØØØ4 .Ø4882 97 19Ø .ØØØ6 81.5 133 16Ø .Ø3587/ PROCESS VALUES
```

Fig. C-4.
Input data to PVGEN--Example 3.

```
(1)    1Ø/           NPV - NUMBER OF PROCESS VARIABLES
(2)    25/           NINV - NUMBER OF VALUES PER INVENTORY PROCESS VARIABLE
(3)    515/          NTRN - NUMBER OF VALUES PER TRANSFER PROCESS VARIABLE
(4)    -12 -12 -12 Ø Ø -24 -24 Ø -1Ø -1Ø/    IPVTRN - TRANSFER INDICATOR
(5)    12.6 .22 .Ø22 .1 .9 9 .15148 .22 35 .ØØØ8/    PROCESS VARIABLES
```

Fig. C-5.
Input data to PVGEN--Example 4.

```
(1)    5/            NPV - NUMBER OF PROCESS VARIABLES
(2)    Ø/            NINV - NUMBER OF VALUES PER INVENTORY PROCESS VARIABLE
(3)    115/          NTRN - NUMBER OF VALUES PER TRANSFER PROCESS VARIABLE
(4)    1 1 1 1 1/    IPVTRN - TRANSFER INDICATOR ARRAY (1 INDICATES TRANSFER P.V.)
(5)    .676 25. .5 75. .22Ø827/    C - PROCESS VARIABLE VALUES
```

Fig. C-6.
Input data to PVGEN--Example 5.

APPENDIX D

MEASIM CODE* LISTING

A complete FORTRAN-IV listing of the MEASIM code as it is used on the PRIME 750 computer is given in Fig. D-1. As indicated in Sec. IX, except for the FORTRAN PARAMETER statement, the code employs very basic FORTRAN. This should make it easily adaptable to other computers.

---

*Los Alamos Identification No. LP 1351.

```
C ***************************** MEASIM *****************************************
C
C*********** MEASUREMENT CODE FOR MODIFIED COPRECAL MODEL ********************
C
C
C
       PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
C    PARAMETER DEFINITIONS:
C        NBALMX - MAXIMUM NUMBER OF BALANCES
C        NPVMX   - MAXIMUM NUMBER OF PV VARIABLES
C        NCIMX - MAXIMUM NUMBER OF INVENTORY COMPONENTS
C        NCTMX - MAXIMUM NUMBER OF TRANSFER COMPONENTS
C        NMIMX - MAXIMUM NUMBER OF INVENTORY MEASUREMENTS
C        NMTMX - MAXIMUM NUMBER OF TRANSFER MEASUREMENTS
C        NSIMX - MAXIMUM NUMBER OF INVENTORY SYSTEMATIC ERRORS
C        NSTMX - MAXIMUM NUMBER OF TRANSFER SYSTEMATIC ERRORS
C        NCMX  - MINIMUM OF NMIMX AND NMTMX
C        MXPMX - MAXIMUM OF NSIMX AND NSTMX
C        NSTRMX - MAXIMUM NUMBER OF RANDOM NUMBER STREAMS
C        NCOLMX - MAXIMUM NUMBER OF PULSE COLUMNS
C
C
       INTEGER*4 ISEED,LSEED
C
C
       DOUBLE PRECISION SUM,SUMSQ
C
       DIMENSION ITIME(4)
C
       INTEGER*2 PVARA, DCNLIN
C
       COMMON /SAMPLE/ SUM(NBMXP1), SUMSQ(NBMXP1)
       COMMON /SEED/ ISEED(NSTRMX), LSEED(NSTRMX)
       COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
       COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
       COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
C
       COMMON /TITLE/ ITITLE(40),IPVTI(30,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
C
C ************** OPEN FILES ******************
C
C          TAPE11  - INPUT DATA
C          TAPE12  - INPUT PV ARRAY
C          TAPE13  - RANDOM NUMBER STARTERS
C          TAPE14  - OUTPUT TO DECANAL
C          TAPE15 - GENERAL OUTPUT FILE
C
      PRINT 10
   10 FORMAT('ENTER INPUT FILE NAMES (ONE PER LINE) FOR'/
     1 '            INPUT DATA'/'           INPUT PV ARRAY'
```

Fig. D-1.
FORTRAN listing of MEASIM.

```
     1 /'            MEASUREMENT OUTPUT TO DECANAL')
      READ(1,2Ø)(MESINP(I),I=1,3)
      READ(1,2Ø)(PVARA(I),I=1,3)
      READ(1,2Ø)(DCNLIN(I),I=1,3)
   2Ø FORMAT(3A2)
      CALL SRCH$$(5,DCNLIN,6,1Ø,Ø,Ø)
      CALL SRCH$$(5,'MESOUT',6,11,Ø,Ø)
      CALL SRCH$$(1,MESINP,6,7,Ø,Ø)
      CALL SRCH$$(1,PVARA,6,8,Ø,Ø)
      CALL SRCH$$(3,'RNSMES',6,9,Ø,Ø)
      CALL SRCH$$(2,DCNLIN,6,1Ø,Ø,Ø)
      CALL SRCH$$(2,'MESOUT',6,11,Ø,Ø)
C     TRACE 3ØØ
C
C ********** SET FILE NUMBER ************
C
      NDAT=11
      NPVIN=12
      NFSEED=13
      NDECIN=14
      NPROUT=15
      NLR=NPROUT-4
      CALL ATTDEV(NPROUT,8,NLR,66)
      CPTIMI=CTIM$A(IDUM)
      IBLANK=
      WRITE(NPROUT,3Ø)(MESINP(I),I=1,3),(PVARA(I),I=1,3),(DCNLIN(I),I=1,
     + 3)
   3Ø FORMAT(1H1,'INPUT DATA FILE - '3A2/' PV ARRAY FILE - '3A2/
     1 ' OUTPUT DECANAL FILE - '3A2/)
C
C ************ READ INPUT DATA **************
C
      CALL MESIN
C
C *********** BEGIN MEASUREMENTS *************
C
   4Ø NRUNA=NRUN
      IF(NRUN.EQ.1) GO TO 7Ø
      WRITE(1,5Ø) ISPNTI
   5Ø FORMAT(//'*******************************************************'//
     1 '            BEGIN MONTE CARLO ISOLATION RUN ON UNIT PROCESS 'I2//
     2 '*******************************************************'//)
      NRUNA=NRUN+1
      DO 6Ø I=1,NBALP1
      SUM(I)=Ø.DØ
   6Ø SUMSQ(I)=Ø.DØ
   7Ø CONTINUE
      DO 11Ø IRUN=1,NRUNA
      WRITE(1,8Ø) IRUN
   8Ø FORMAT(' RUN NUMBER = 'I4)
      IF(NRUN.EQ.1) GO TO 9Ø
      IZE=Ø
      IF(IRUN.EQ.1) IZE=1
   9Ø CALL MESDRV
      DO 1ØØ I=1,NNSTRM
      ISEED(I)=LSEED(I)
  1ØØ CONTINUE
  11Ø CONTINUE
      IF(NRUN.GT.1)CALL STNDEV
      READ(NDAT,*,END=13Ø) ISPNTI
      CALL SETMAS
      REWIND NFSEED
      DO 12Ø I=1,NNSTRM
      READ(NFSEED,*)ISEED(I)
      LSEED(I)=ISEED(I)
```

Fig. D-1. (cont)

```
  12Ø CONTINUE
      GO TO 4Ø
  13Ø CONTINUE
C
C ***************** SAVE LAST RANDOM NUMBER SEED *****************
C
      IF(IRNSCH.EQ.Ø)GO TO 16Ø
      REWIND NFSEED
      DO 14Ø I=1,NNSTRM
      WRITE(NFSEED,15Ø)LSEED(I)
  14Ø CONTINUE
  15Ø FORMAT(I1Ø)
  16Ø CONTINUE
      WRITE(NPROUT,17Ø)(LSEED(I),I=1,NNSTRM)
  17Ø FORMAT(1H1' FINAL RANDOM NUMBER SEEDS'/(1ØI12))
      CPTIMF=CTIMSA(IDUM)
      CPTIME=CPTIMF-CPTIMI
      WRITE(NPROUT,18Ø) CPTIME
  18Ø FORMAT(//,' CPU TIME = 'F12.6//)
      CALL TIMESA(ITIME)
      WRITE(NPROUT,19Ø) (ITIME(I),I=1,4)
  19Ø FORMAT(//1X,4A2)
      CALL CLOSEM
      CALL EXIT
      END
      SUBROUTINE BLANKS(N)
C
C *********** READS BLANK LINES TO BE USED FOR INPUT DATA COMMENTING PURPOSES
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DO 1Ø I=1,N
      READ(NDAT,*)NBLANK
   1Ø CONTINUE
      RETURN
      END
      SUBROUTINE CLOSEM
C
C ************** CLOSES FILES AND TERMINATES RUN
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
      INTEGER*2 DCNLIN, PVARA
C
      COMMON /TITLE/ ITITLE(4Ø),IPVTI(3Ø,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
C
C ****************** CLOSE FILES *******************
```

Fig. D-1. (cont)

```
C
      CALL SRCHS$(4,MESINP,6,7,Ø,Ø)
      CALL SRCHS$(4,PVARA,6,8,Ø,Ø)
      CALL SRCHS$(4,'RNSMES',6,9,Ø,Ø)
      CALL SRCHS$(4,DCNLIN,6,1Ø,Ø,Ø)
      CALL SRCHS$(4,'MESOUT',6,11,Ø,Ø)
      CALL EXIT
      RETURN
      END
      SUBROUTINE COLUMN(IPV,IC,IT)
C
C         COMPUTES INVENTORY COMPONENTS FOR THE COLUMNS.
C
C ****** THE GENERAL COLUMN INVENTORY EQUATION IS OF THE FORM
C
C         XI=CF*CON(1)+CW*CON(2)+CP*CON(3)+VORGT*CON(4)+VAQB*CON(5)+CON(6)
C
C           WHERE:
C            CF              = FEED CONCENTRATION
C            CW              = WASTE CONCENTRATION
C            CP              = PRODUCT CONCENTRATION
C            VORGT           = VOLUME OF ORGANIC TOP SECTION
C            VAQB            = VOLUME OF AQUEOUS BOTTOM SECTION
C            CON(1)-CON(6) = CONSTANTS FOR A SPECIFIC COLUMN
C
C        THE PROCESS VARIABLES ARE CF,CW,CP,VORGT, AND VAQB.
C        THE MEASURED VALUES ARE CF,CW, AND CP
C
C
C        CALLING ARGUMENTS:
C           IPV  - PROCESS VARIABLE NUMBER FOR FIRST PROCESS VARIABLE (CF)
C           IC   - COLUMN NUMBER
C           IT   - COLUMN TYPE
C                  .EQ. 1 - "A" TYPE COLUMN
C                  .EQ. 2 - "B" TYPE COLUMN
C                  .EQ. 3 - "S" TYPE COLUMN
C                  .EQ. 4 - "Z" TYPE COLUMN
C
C
       PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
       COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
       COMMON /PVCOMS/ PVIS(NBMXP3,NPVIMX),PVTS(NTMXP2,NPVTMX)
C
       COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
       COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
C
       DIMENSION CON(6), IPV(1)
C
```

Fig. D-1. (cont)

```
      COMMON /WORK/ X(NTMXP2,3)
C
C *************** COMPUTE MEASURED VALUES FOR CF, CW, AND CP
C                        X(I,1) = CF(I)
C                        X(I,2) = CW(I)
C                        X(I,3) = CP(I)
C
      ICINVS=ICINV
      DO 40 J=1,3
      IPVJ=IPV(J)
      IPVTI=NPVIT(IPVJ)
      IF(IPVTRN(IPVJ).EQ.0)GO TO 20
      DO 10 I=1,NBALP2
      ITC=1+(I-1)*NTRPBL
      X(I,J)=PVT(ITC,IPVTI)
   10 CONTINUE
      GO TO 40
   20 DO 30 I=1,NBALP2
      X(I,J)=PVI(I,IPVTI)
   30 CONTINUE
   40 CONTINUE
C
C *************** COMPUTE THE COLUMN CONSTANTS
C
      GO TO (50,60,50,60),IT
   50 CONTINUE
C
C *************** TYPE "A" OR "S" COLUMNS
C
      CON(1)=(HCC(1,IC)+HCC(4,IC))/CCC(3,IC)
      CON(2)=VCC(2,IC)
      CON(3)=(HCC(2,IC)+HCC(3,IC)+HCC(5,IC))/CCC(1,IC)
      CON(4)=HCC(3,IC)/VCC(1,IC)
      CON(5)=CCC(2,IC)
      CON(6)=-VCC(2,IC)*CCC(2,IC)-HCC(3,IC)
      GO TO 70
   60 CONTINUE
C
C *************** TYPE "B" OR "Z" COLUMNS
C
      CON(1)=(HCC(2,IC)+HCC(5,IC))/CCC(3,IC)
      CON(2)=VCC(1,IC)
      CON(3)=(HCC(1,IC)+HCC(3,IC)+HCC(4,IC))/CCC(1,IC)
      CON(4)=CCC(2,IC)
      CON(5)=HCC(3,IC)/VCC(2,IC)
      CON(6)=-VCC(1,IC)*CCC(2,IC)-HCC(3,IC)
   70 CONTINUE
C
      DO 80 J=1,3
      ICINV=ICINV+1
      DO 80 I=1,NBALP2
      XIM(I,ICINV,1)=X(I,J)*CON(J)
      XI(I,ICINV)=XIM(I,ICINV,1)
   80 CONTINUE
      IF(ICINV.LE.NCITST)GO TO 100
      WRITE(1,90) IPV
      WRITE(NPROUT,90) IPV
   90 FORMAT(//' RUN TERMINATED IN SUBROUTINE COLUMN WITH ICINV .GT. NCI
     1TST'/'    PROCESS VARIABLE NUMBER ='I3)
      CALL CLOSEM
  100 CONTINUE
C
C *************** ADD NON-MEASURED TERMS TO THE LAST INVENTORY COMPONENT
C
      IPVTI4=NPVIT(IPV(4))
```

Fig. D-1. (cont)

```
      IPVTI5=NPVIT(IPV(5))
      DO 110 I=1,NBALP2
      XI(I,NCI)=XI(I,NCI)+PVI(I,IPVTI4)*CON(4)+PVI(I,IPVTI5)*CON(5)
     +    + CON(6)
      XIM(I,NCI,1)=XI(I,NCI)
  110 CONTINUE
      IF(NMT.EQ.1) GO TO 130
      JS=ICINVS+1
      DO 120 I=1,NBALP2
      XIM(I,NCI,2)=1.
      DO 120 J=JS,ICINV
      XIM(I,J,2)=1.
  120 CONTINUE
C
  130 RETURN
      END
      SUBROUTINE CUSUM
C
C ***********************************************************************
C                 COMPUTES CUSUM AND CUSUM VARIANCE
C ***********************************************************************
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
C
      DOUBLE PRECISION SUM,SUMSQ,DCS
C
      COMMON /SAMPLE/ SUM(NBMXP1), SUMSQ(NBMXP1)
C
      INTEGER TSCI,TCI
C
      COMMON /MSINCM/ ICI(NCIMX,NCIMX,NMIMX), ISCI(NCIMX,NMIMX),
     + ITCI(NCIMX,NCTMX,NCMX), TCI(NCTMX,NCTMX,NMTMX),TSCI(NCTMX,NMTMX),
     +  VIR(NCIMX,NMIMX), VTR(NCTMX,NMTMX),VIS(NCIMX,NMIMX,2),
     +  VTS(NCTMX,NMTMX,2), SIR(NCIMX,NMIMX),STR(NCTMX,NMTMX),
     +  SIS(NCIMX,NMIMX,2), STS(NCTMX,NMTMX,2)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
C
      DIMENSION JS(NCTMX,NCTMX,NMTMX,NSTMX)
C
C
C ***********************************************************************
C
C           CS     - CUSUM
C           S2CS   - VARIANCE OF CUSUM
C           CVT    - COVARIANCE BETWEEN TRANSFERS
C
C ***********************************************************************
```

Fig. D-1. (cont)

```
C
C
C *************** COMPUTE INVENTORY AND TRANSFER VARIANCES AND COVARIANCES
C
      IF(IRUN.EQ.1)CALL MASCUS
C
C ************** DOWNSHIFT TRANSFER ARRAYS FOR COMPATIBILITY WITH
C                THE MESSIM CUSUM ALGORITHM
C
      NTRNM1=NTRN-1
      DO 1Ø I=1,NTRNM1
      II=NTRN+1-I
      IIM1=II-1
      TT(II)=TT(IIM1)
      S2T(II)=S2T(IIM1)
      CVT(II)=CVT(IIM1)
   1Ø CONTINUE
      TT(1)=Ø.
      TTSUM(1)=Ø.
      S2T(1)=Ø.
      S2TRSM(1)=Ø.
      S2TBSM(1)=Ø.
      CVT(1)=Ø.
      CS(1)=Ø
      S2CS(1)=2.ØEØ*S2I(1)
      S2CST=S2CS(1)
C
C *************** INITIALIZE JS ARRAY
C
      DO 2Ø II=1,NCT
      DO 2Ø J=1,NCT
      DO 2Ø K=1,NMT
      DO 2Ø L=1,NST
      JS(II,J,K,L)=1
   2Ø CONTINUE
      IRC=1
      IT=1
      DO 22Ø I=2,NBALP1
      TTSUM(I)=Ø.
      S2TRSM(I)=Ø.
      S2TBSM(I)=Ø.
      DO 3Ø II=1,NTRPBL
      ITM1=IT
      IT=IT+1
      TTSUM(I)=TTSUM(I)+TT(IT)
      S2TRSM(I)=S2TRSM(I)+S2TR(ITM1)
      S2TBSM(I)=S2TBSM(I)+S2TB(ITM1)
   3Ø CONTINUE
      CVX=Ø.
C
C *************** COMPUTE CORRELATIONS BETWEEN TRANSFERS IN SAME
C ***************    BALANCE PERIOD
C
      IF(NTRPBL.EQ.1)GO TO 1ØØ
      IR1=(I-2)*NTRPBL+2
      JST=IR1-1
      IR2=(I-1)*NTRPBL
      DO 9Ø IR=IR1,IR2
      JF=IR-1
      DO 8Ø KK=1,NCT
      IF(T(IR,KK).EQ.Ø)GO TO 8Ø
      DO 7Ø K=1,NCT
      DO 6Ø L=1,NMT
      DO 5Ø M=1,NST
      IF(TCI(K,KK,L).LT.M)GO TO 5Ø
```

Fig. D-1. (cont)

```
      DO 4Ø J=JST,JF
      IF(T(J,K).EQ.Ø)GO TO 4Ø
      CALL MTFIX(4,J,K,L,M,XX1)
      CALL MTFIX(4,IR,KK,L,M,XX2)
      CVX=CVX+XX1*XX2*STS(K,L,M)*STS(KK,L,M)
   4Ø CONTINUE
   5Ø CONTINUE
   6Ø CONTINUE
   7Ø CONTINUE
   8Ø CONTINUE
   9Ø CONTINUE
      S2TBSM(I)=S2TBSM(I)+2.*CVX
  1ØØ CONTINUE
      IM1=I-1
      CS(I)=CS(IM1)+XIT(IM1)-XIT(I)+TTSUM(I)
      IF(IRUN.EQ.1) GO TO 11Ø
      DCS=DBLE(CS(I))
      SUM(I)=SUM(I)+DCS
      SUMSQ(I)=SUMSQ(I)+DCS**2
      GO TO 22Ø
  11Ø CONTINUE
C
C *************** CUSUM VARIANCE
C
      S2CST=S2CST-S2I(I-1)+S2I(I)+S2TRSM(I)+S2TBSM(I)
C
C *************** TRANSFER COVARIANCE
C
      DO 18Ø II=1,NTRPBL
      IRC=IRC+1
      IRCM1=IRC-1
      IF(I.EQ.2)GO TO 18Ø
      IF(IRC.LT.3) GO TO 18Ø
      JF=(I-2)*NTRPBL
      TCV=Ø.
      DO 17Ø KK=1,NCT
      IF(T(IRCM1,KK).EQ.Ø)GO TO 17Ø
      DO 16Ø K=1,NCT
      DO 15Ø L=1,NMT
      DO 14Ø M=1,NST
      IF(TCI(K,KK,L).LT.M)GO TO 14Ø
      JST=JS(K,KK,L,M)
      JINT=JF-JST+1
      IF(JINT.LT.NCAL(K,L,M))GO TO 12Ø
      JS(K,KK,L,M)=JF+1
      GO TO 14Ø
  12Ø CONTINUE
      IF(JF.LT.JST)GO TO 14Ø
      DO 13Ø J=JST,JF
      IF(T(J,K).EQ.Ø)GO TO 13Ø
      CALL MTFIX(4,J,K,L,M,XX1)
      CALL MTFIX(4,IRCM1,KK,L,M,XX2)
      TCV=TCV+XX1*XX2*STS(K,L,M)*STS(KK,L,M)
  13Ø CONTINUE
  14Ø CONTINUE
  15Ø CONTINUE
  16Ø CONTINUE
  17Ø CONTINUE
      S2CST=S2CST+2.*TCV
  18Ø CONTINUE
C
C *************** INVENTORY COVARIANCE
C
  19Ø XICV=Ø.
      DO 21Ø K=1,NCI
```

Fig. D-1. (cont)

```
      DO 210 KK=1,NCI
      DO 210 L=1,NMI
      DO 200 M=1,NST
      IF(ICI(K,KK,L).LT.M)GO TO 200
      CALL MTFIX(2,1,K,L,M,XX1)
      CALL MTFIX(2,I,KK,L,M,XX2)
      XICV=XICV+XX1*XX2*SIS(K,L,M)*SIS(KK,L,M)
  200 CONTINUE
  210 CONTINUE
C
C *************** TOTAL CUSUM VARIANCE
C
      S2CS(I)=S2CST-2.*XICV
  220 CONTINUE
      IF(IRUN.GT.1)GO TO 300
      S2CS(1)=0.0E0
C
C ************* COVARIANCE BETWEEN ADJACENT TRANSFER BLOCKS
C
      IF(ICLAPS.NE.2)GO TO 300
      DO 290 I=2,NBALP1
      CVT(I)=0.
      IR1=(I-1)*NTRPBL+1
      JST=IR1-NTRPBL
      NNF=IR1-1
      IR2=I*NTRPBL
      DO 280 IR=IR1,IR2
      JF=IR1-1
      DO 270 KK=1,NCT
      IF(T(IR,KK).EQ.0)GO TO 270
      DO 260 K=1,NCT
      DO 250 L=1,NMT
      DO 240 M=1,NST
      IF(TCI(K,KK,L).LT.M)GO TO 240
C
C *************** CHECK FOR RECALIBRATION
C
      IDUM=NCAL(K,L,M)
      IF(MOD(NNF,IDUM).EQ.0)GO TO 240
      DO 230 J=JST,JF
      IF(T(J,K).EQ.0)GO TO 230
      CALL MTFIX(4,J,K,L,M,XX1)
      CALL MTFIX(4,IR,KK,L,M,XX2)
      CVT(I)=CVT(I)+XX1*XX2*STS(K,L,M)*STS(KK,L,M)
  230 CONTINUE
  240 CONTINUE
  250 CONTINUE
  260 CONTINUE
  270 CONTINUE
  280 CONTINUE
  290 CONTINUE
  300 CONTINUE
      RETURN
      END
      FUNCTION DRAND(ISTRM)
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPV1T(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
```

Fig. D-1. (cont)

```
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      INTEGER*4 LSEED,ISEED,M2,ITWO,IC,IA,MIC
      COMMON /SEED/ ISEED(NSTRMX), LSEED(NSTRMX)
C
C     DRAND IS A UNIFORM RANDOM NUMBER GENERATOR BASED ON THEORY AND
C SUGGESTIONS GIVEN IN D.E. KNUTH (1969), VOL 2.  THE INTEGER ISTRM
C SHOULD BE INITIALIZED TO AN ARBITARY INTEGER PRIOR TO THE FIRST CALL
C TO DRAND. THE CALLING PROGRAM SHOULD NOT ALTER THE VALUE OF ISTRM
C BETWEEN SUBSEQUENT CALLS TO DRAND. VALUES OF DRAND WILL BE RETURNED
C IN THE INTERVAL(Ø,1)
C
      DOUBLE PRECISION HALFM
C     DATA M2/1Ø73741824/,ITWO/2/,IA/843314861/,IC/453816693/,
C    $ MIC/169366955/
C     DATA HALFM/.1Ø737418240D+1Ø/,S/.4656613E-Ø9/
      PARAMETER (M2=1Ø73741824, ITWO=2, IA=843314861, IC=453816693,
     + MIC=169366955, HALFM=.1Ø737418240D+1Ø, S=.4656613E-Ø9)
C
      IF(ISTRM.GT.Ø)GO TO 3Ø
      IF(ISTRM.LT.Ø)GO TO 2Ø
      WRITE(1,1Ø)
      WRITE(NPROUT,1Ø)
   1Ø FORMAT(//' YOU HAVE CALLED DRAND WITH ISTRM=Ø'
     1 /' ************ RUN TERMINATED ****************')
      CALL  CLOSEM
   2Ø ISTRG=-ISTRM
      LSEED(ISTRG)=ISEED(ISTRG)
      RETURN
   3Ø ISTRG=ISTRM
C
C COMPUTE NEXT RANDOM NUMBER
C
      LSEED(ISTRG) = LSEED(ISTRG)*IA
      LSEED(ISTRG) = LSEED(ISTRG) + IC
C
C THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE INTEGER
C OVERFLOW AFFECTS THE SIGN BIT
C
      IF(LSEED(ISTRG).LT.Ø) LSEED(ISTRG) = (LSEED(ISTRG) + M2) + M2
      DRAND = FLOAT(LSEED(ISTRG))*S
   4Ø RETURN
      END
      SUBROUTINE INV1(IPV)
C
C ************** INVENTORY MODEL WITH ONE MEASUREMENT
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     $ NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     $ S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
```

Fig. D-1. (cont)

```
      1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
      2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
      3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
      4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
      5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
       DIMENSION IPV(1)
C
C
C
C                PV(1,IPV) = PROCESS VALUE OF HOLDUP
C                IPV       = PV VARIABLE NUMBER
C
C
C
       ICINV=ICINV+1
       IF(ICINV.LE.NCITST)GO TO 20
       WRITE(1,10) IPV(1)
       WRITE(NPROUT,10) IPV(1)
    10 FORMAT(//' RUN TERMINATED IN SUBROUTINE INV1 WITH ICINV .GT. NCITS
      1T'/'     PROCESS VARIABLE NUMBER ='I3)
       CALL CLOSEM
    20 CONTINUE
       IPV1=IPV(1)
       IPVTI=NPVIT(IPV1)
       IF(IPVTRN(IPV1).EQ.0)GO TO 40
       DO 30 I=1,NBALP2
       ITC=1+(I-1)*NTRPBL
       XIM(I,ICINV,1)=PVT(ITC,IPVTI)
    30 CONTINUE
       GO TO 60
    40 DO 50 I=1,NBALP2
       XIM(I,ICINV,1)=PVI(I,IPVTI)
    50 CONTINUE
C
    60 DO 70 I=1,NBALP2
       XI(I,ICINV)=XIM(I,ICINV,1)
    70 CONTINUE
       IF(NMI.EQ.1)GO TO 90
       DO 80 I=1,NBALP2
       XIM(I,ICINV,2)=1.
    80 CONTINUE
C
    90 RETURN
       END
       SUBROUTINE INV2(IPV)
C
C ************* INVENTORY MODEL WITH TWO MEASUREMENTS
C
       PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
      1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
      2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
      2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
      4 NCOLMX=2)
C
       COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
       COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
      + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
      + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
      + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
      + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
       COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
      $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
      1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
```

Fig. D-1. (cont)

```
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DIMENSION IPV(1)
C
C
C
C              PV(I,IPV) = PROCESS VALUE OF FIRST MEASUREMENT
C              PV(I,IPV+1) = PROCESS VALUE OF SECOND MEASUREMENT
C
C
      ICINV=ICINV+1
      IF(ICINV.LE.NCITST)GO TO 2Ø
      WRITE(1,1Ø) IPV(1)
      WRITE(NPROUT,1Ø) IPV(1)
   1Ø FORMAT(//' RUN TERMINATED IN SUBROUTINE INV2 WITH ICINV .GT. NCITS
     1T'/'    PROCESS VARIABLE NUMBER ='I3)
      CALL CLOSEM
   2Ø CONTINUE
      DO 6Ø J=1,2
      IPVJ=IPV(J)
      IPVTI=NPVIT(IPVJ)
      IF(IPVTRN(IPVJ).EQ.Ø)GO TO 4Ø
      DO 3Ø I=1,NBALP2
      ITC=1+(I-1)*NTRPBL
      XIM(I,ICINV,J)=PVT(ITC,IPVTI)
   3Ø CONTINUE
      GO TO 6Ø
   4Ø CONTINUE
      DO 5Ø I=1,NBALP2
      XIM(I,ICINV,J)=PVI(I,IPVTI)
   5Ø CONTINUE
   6Ø CONTINUE
C
      DO 7Ø I=1,NBALP2
      XI(I,ICINV)=XIM(I,ICINV,1)*XIM(I,ICINV,2)
   7Ø CONTINUE
C
   8Ø RETURN
      END
      SUBROUTINE MASAGE(IB,IT,III)
C
C *************** COMPUTES VARIANCES AND COVARIANCES FOR INDIVIDUAL
C *************** TRANSFER AND INVENTORY COMPONENTS
C
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
C   PARAMETER DEFINITIONS:
C       NBALMX - MAXIMUM NUMBER OF BALANCES
C       NPVMX   - MAXIMUM NUMBER OF PV VARIABLES
C       NCIMX - MAXIMUM NUMBER OF INVENTORY COMPONENTS
C       NCTMX - MAXIMUM NUMBER OF TRANSFER COMPONENTS
C       NMIMX - MAXIMUM NUMBER OF INVENTORY MEASUREMENTS
C       NMTMX - MAXIMUM NUMBER OF TRANSFER MEASUREMENTS
C       NSIMX - MAXIMUM NUMBER OF INVENTORY SYSTEMATIC ERRORS
C       NSTMX - MAXIMUM NUMBER OF TRANSFER SYSTEMATIC ERRORS
C       NCMX  - MINIMUM OF NMIMX AND NMTMX
C       MXPMX - MAXIMUM OF NSIMX AND NSTMX
```

Fig. D-1. (cont)

```
C        NSTRMX - MAXIMUM NUMBER OF RANDOM NUMBER STREAMS
C        NCOLMX - MAXIMUM NUMBER OF PULSE COLUMNS
C
      COMMON /CVCOM/ CVSIS(NCIMX,NMIMX,NSIMX), CVITS(NCIMX,NCTMX,NMTMX,
     + MXPMX), CVTS(NCTMX,NCTMX,NMTMX,NSTMX), CVSTS(NCTMX,NMTMX,NSTMX),
     + CVIS(NCIMX,NCIMX,NMIMX,NSIMX), CVITR(NCIMX,NCTMX,NMTMX),VTIR,
     + XCVSIS(NCIMX,NCIMX,NMIMX,NSIMX), XCVSTS(NCTMX,NCTMX,NMTMX,NSTMX),
     + VTTR, XCVITS(NCIMX,NCTMX,NMTMX,MXPMX)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      INTEGER TSCI,TCI
C
      COMMON /MSINCM/ ICI(NCIMX,NCIMX,NMIMX), ISCI(NCIMX,NMIMX),
     + ITCI(NCIMX,NCTMX,NCMX), TCI(NCTMX,NCTMX,NMTMX),TSCI(NCTMX,NMTMX),
     +  VIR(NCIMX,NMIMX), VTR(NCTMX,NMTMX),VIS(NCIMX,NMIMX,2),
     +  VTS(NCTMX,NMTMX,2), SIR(NCIMX,NMIMX),STR(NCTMX,NMTMX),
     +  SIS(NCIMX,NMIMX,2), STS(NCTMX,NMTMX,2)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
C
      DIMENSION IREC(6,2,2)
C
      INTEGER*2 DCNLIN, PVARA
C
      COMMON /TITLE/ ITITLE(4Ø),IPVTI(3Ø,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
      DATA IREC/2,1,1,1,3,3,4,3,2,4,2,4,2,1,1,1,4,4,3,4,2,3,2,3/
C
C           VARIABLE NAMES
C              INPUT
C         NCI  NUMBER OF INVENTORY COMPONENTS
C         NCT  NUMBER OF TRANSFER COMPONENTS
C         NMI  NUMBER OF INVENTORY MEASUREMENTS
C         NMT  NUMBER OF TRANSFER MEASUREMENTS
C           EXAMPLE: LET H=C1*V1+C2*V2 THEN C1*V1 IS FIRST COMPONENT
C           AND C2 IS THE FIRST MEASUREMENT VALUE OF SECOND COMPONENT
C         NSI  NUMBER OF ETA(BIAS) TERMS IN EACH INVENTORY MEASUREMENT
C         NST  NUMBER OF ETA(BIAS) TERMS IN EACH TRANSFER MEASUREMENT
C          NC  MINIMUM OF NMI AND NMT
C         NP  TOTAL NUMBER OF BALANCE PERIODS
C         NNP  NUMBER OF CURRENT BALANCE PERIODS
C           I COMPONENT INDEX
C           I1 COMPONENT INDEX
C           J VALUE INDEX
C           K BIAS TERM INDEX
C         XI(I,J)  INVENTORY MEASUREMENT COMPONENT
C         T(I,J)  TRANSFER MEASUREMENT COMPONENT
C         VIR(I,J)  VARIANCE OF INVENTORY RANDOM ERROR
C         VTR(I,J)  VARIANCE OF TRANSFER RANDOM ERROR
C         VIS(I,J,K)  VARIANCE OF INVENTORY BIAS ERROR
C         VTS(I,J,K)  VARIANCE OF TRANSFER BIAS ERROR
C         ICI(I,I1,J)  INVENTORY CORRELATION INDICATOR
```

Fig. D-1. (cont)

```
C          TCI(I,I1,J)  TRANSFER CORRELATION INDICATOR
C          ITCI(I,I1,J)  INVENTORY-TRANSFER CORRELATION INDICATOR
C          ISCI(I,J)  INVENTORY SEQUENTIAL CORRELATION INDICATOR
C          TSCI(I,J)  TRANSFER SEQUENTIAL CORRELATION INDICATOR
C            NOTE: CORRELATION INDICATORS ARE INTEGERS
C              OUTPUT
C          XINVT  TOTAL INVENTORY MEASUREMENT
C          TR  TOTAL TRANSFER MEASUREMENT
C          CVIS(I,I1,J,K)  COVARIANCE OF INVENTORY SYSTEMATIC ERRORS
C          CVTS(I,I1,J,K)  COVARIANCE OF TRANSFER SYSTEMATIC ERRORS
C          VTIR  VARIANCE OF TOTAL INVENTORY RANDOM ERRORS
C          VTTR  COVARIANCE OF TOTAL TRANSFER RANDOM ERRORS
C          CVSIS(I,J,K)  COVARIANCE BETWEEN SUCCESSIVE INVENTORY SYSTEMATIC ERRORS
C          CVSTS(I,J,K)  COVARIANCE BETWEEN SUCCESSIVE TRANSFER SYSTEMATIC ERRORS
C          CVITR(I,I1,J,K)  COVARIANCE BETWEEN INVENTORY-TRANSFER RANDOM ERRORS
C          CVITS(I,I1,J,K)  COVARIANCE BETWEEN INVENTORY-TRANSFER SYSTEMATIC ERROR
C
C      TRACE 999
C
C
       IF(IB.NE.1)GO TO 70
       IF(IT.NE.1)GO TO 70
    10 FORMAT(3A2)
    20 FORMAT(40A2)
       WRITE(NDECIN,20)(ITITLE(I),I=1,40)
       WRITE(NDECIN,10) IBLANK
       WRITE(NDECIN,10) IBLANK
       WRITE(NDECIN,10) IBLANK
       IGO=ICLAPS+1
       GO TO (30,40,50),IGO
    30 CONTINUE
C
C *************** NORMAL DECANAL INPUT (NO COLLAPSING)
C
       NCIP=NCI
       NCTP=NCT
       NMIP=NM1
       NMTP=NMT
       NSIP=NSI
       NSTP=NST
       NCP=NC
       GO TO 60
    40 CONTINUE
C
C *************** COLLAPSE INVENTORY - NO COLLAPSE OF TRANSFER
C
       NCIP=1
       NCTP=NCT
       NMIP=1
       NMTP=NMT
       NSIP=1
       NSTP=NST
       NCP=1
       GO TO 60
    50 CONTINUE
C
C *************** COLLAPSE BOTH TRANSFERS AND INVENTORIES
C
       NCIP=1
       NCTP=1
       NMIP=1
       NMTP=1
       NSIP=1
       NSTP=1
       NCP=1
```

Fig. D-1. (cont)

```
   6Ø CONTINUE
      NP=NBALP1-1
      WRITE(NDECIN,53Ø)NCIP,NCTP,NMIP,NMTP,NSIP,NSTP,NCP,NP
      WRITE(NDECIN,9Ø) ICLAPS
   7Ø CONTINUE
      IF(ICLAPS.GT.Ø)GO TO 1ØØ
      IF(III.EQ.1)WRITE(NDECIN,8Ø)((XIM(IB,I,J),J=1,NMI),I=1,NCI)
      WRITE(NDECIN,8Ø)((TM(IB,I,J),J=1,NMT),I=1,NCT)
      WRITE(NDECIN,9Ø)IT
      IF(IB.NE.1)GO TO 1ØØ
      IF(IT.NE.1)GO TO 1ØØ
C
C ************* CONVERT VARIANCES TO STANDARD DEVIATIONS FOR INPUT
C ************* TO DECANAL
C
      WRITE(NDECIN,8Ø)((SIR(I,J),J=1,NMI),I=1,NCI)
      WRITE(NDECIN,8Ø)((STR(I,J),J=1,NMT),I=1,NCT)
      WRITE(NDECIN,8Ø)(((SIS(I,J,K),K=1,NSI),J=1,NMI),I=1,NCI)
      WRITE(NDECIN,8Ø)(((STS(I,J,K),K=1,NST),J=1,NMT),I=1,NCT)
      WRITE(NDECIN,9Ø)(((ICI(I,I1,J),J=1,NMI),I1=1,NCI),I=1,NCI)
      WRITE(NDECIN,9Ø)(((TCI(I,I1,J),J=1,NMT),I1=1,NCT),I=1,NCT)
      WRITE(NDECIN,9Ø)(((ITCI(I,I1,J),J=1,NC),I1=1,NCT),I=1,NCI)
      WRITE(NDECIN,9Ø)((ISCI(I,J),J=1,NMI),I=1,NCI)
      WRITE(NDECIN,9Ø)((TSCI(I,J),J=1,NMT),I=1,NCT)
   8Ø FORMAT(8G15.7)
   9Ø FORMAT(2ØI4)
  1ØØ CONTINUE
C
      ITM1=IT-1
      ITP1=IT+1
      IF(III.NE.1)GO TO 19Ø
      IBP1=IB+1
C
C          INVENTORY MEASUREMENTS
C
      XINVT=Ø.
      DO 11Ø I=1,NCI
  11Ø XINVT=X1(IB,I)+XINVT
C
C          VARIANCE OF TOTAL RANDOM INVENTORY ERROR
      VTIR=Ø.
      DO 12Ø I=1,NCI
      DO 12Ø J=1,NMI
      CALL MTFIX(1,IB,I,J,IDUM,XX1)
  12Ø VTIR=VTIR+XX1**2*VIR(I,J)
C
C          COVARIANCE OF INVENTORY SYSTEMATIC ERRORS
      DO 14Ø I=1,NCI
      DO 14Ø I1=1,NCI
      DO 14Ø J=1,NMI
      KF=ICI(I,I1,J)
      IF(KF.EQ.Ø)GO TO 14Ø
      DO 13Ø K=1,KF
      CALL MTFIX(2,IB,I,J,K,XX1)
      CALL MTFIX(2,IB,I1,J,K,XX2)
      CVIS(I,I1,J,K)=XX1*XX2*SIS(I,J,K)*SIS(I1,J,K)
  13Ø CONTINUE
  14Ø CONTINUE
C
C          COVARIANCE BETWEEN SUCCESSIVE INVENTORY SYSTEMATIC ERRORS
      DO 16Ø I=1,NCI
      DO 16Ø J=1,NMI
      KF=ISCI(I,J)
      IF(KF.EQ.Ø)GO TO 16Ø
      DO 15Ø K=1,KF
```

Fig. D-1. (cont)

```
      CALL MTFIX(2,IBP1,I,J,K,XX1)
      CALL MTFIX(2,IB,I,J,K,XX2)
      CVSIS(I,J,K)=XX1*XX2*VIS(I,J,K)
  15Ø CONTINUE
  16Ø CONTINUE
C
      DO 18Ø I=1,NCI
      DO 18Ø J=1,NMI
      DO 18Ø K=1,NSI
      IF(CVSIS(I,J,K) .LT. 1.E-Ø9) GO TO 18Ø
      DO 17Ø I1=1,NCI
      IF(CVIS(I,I1,J,K) .LT. 1.E-Ø9) GO TO 17Ø
      CALL MTFIX(2,IBP1,I1,J,K,XX1)
      CALL MTFIX(2,IB,I,J,K,XX2)
      XCVSIS(I,I1,J,K)=XX1*XX2*SIS(I,J,K)*SIS(I1,J,K)
  17Ø CONTINUE
  18Ø CONTINUE
C
C           TRANSFER MEASUREMENTS
  19Ø TR=Ø.
      DO 2ØØ I=1,NCT
  2ØØ TR=T(IB,I)+TR
C
C           VARIANCE OF TOTAL RANDOM TRANSFER ERROR
      VTTR=Ø.
      DO 21Ø I=1,NCT
      DO 21Ø J=1,NMT
      CALL MTFIX(3,IT,I,J,IDUM,XX1)
  21Ø VTTR=VTTR+XX1**2*VTR(I,J)
C
C ************* RANDOM ERROR CORRELATIONS
C ************* DEVELOPED FOR THE CASE WHERE TRANSFER IS COMPUTED AS
C ************* THE INTEGRATED PRODUCT OF FLOW RATE AND CONCENTRATION
C
      IF(NCFRC.EQ.Ø)GO TO 25Ø
      DO 24Ø II=1,NCFRC
      DO 24Ø K=1,2
      DO 22Ø I=1,2
      NCOMP1=ICFR(II)+IREC(I,1,K)-1
      NCOMP2=ICFR(II)+IREC(I,2,K)-1
      CALL MTFIX(3,IT,NCOMP1,K,IDUM,XX1)
      CALL MTFIX(3,IT,NCOMP2,K,IDUM,XX2)
      VTTR=VTTR+2.*XX1*XX2*STR(NCOMP1,K)*STR(NCOMP2,K)
  22Ø CONTINUE
      IF(IT.EQ.1)GO TO 24Ø
      DO 23Ø I=3,6
      NCOMP1=ICFR(II)+IREC(I,1,K)-1
      NCOMP2=ICFR(II)+IREC(I,2,K)-1
      CALL MTFIX(3,IT,NCOMP1,K,IDUM,XX1)
      CALL MTFIX(3,ITM1,NCOMP2,K,IDUM,XX2)
      VTTR=VTTR+2.*XX1*XX2*STR(NCOMP1,K)*STR(NCOMP2,K)
  23Ø CONTINUE
  24Ø CONTINUE
  25Ø CONTINUE
C
C ********* RANDOM ERROR CORRELATIONS
C ********* DEVELOPED FOR THE CASE WHERE THE TRANSFER IS COMPUTED
C ********* THE PRODUCT OF CONCENTRATION AND WEIGHT CHANGE
C
      IF(NCGF.EQ.Ø)GO TO 27Ø
      DO 26Ø II=1,NCGF
      NCOMP1=ICGF(II)
      NCOMP2=ICGF(II)+1
      CALL MTFIX(3,IT,NCOMP1,1,IDUM,XX1)
      CALL MTFIX(3,IT,NCOMP2,1,IDUM,XX2)
```

Fig. D-1. (cont)

```
      VTTR=VTTR+2.*XX1*XX2*STR(NCOMP1,1)*STR(NCOMP2,1)
  260 CONTINUE
  270 CONTINUE
C
C           COVARIANCE OF TRANSFER SYSTEMATIC ERRORS
      DO 290 I=1,NCT
      DO 290 I1=1,NCT
      DO 290 J=1,NMT
      KF=TCI(I,I1,J)
      IF(KF.EQ.0)GO TO 290
      DO 280 K=1,KF
      CALL MTFIX(4,IT,I,J,K,XX1)
      CALL MTFIX(4,IT,I1,J,K,XX2)
      CVTS(I,I1,J,K)=XX1*XX2*STS(I,J,K)*STS(I1,J,K)
  280 CONTINUE
  290 CONTINUE
      IF(INTRNC.EQ.0)GO TO 340
C
C           COVARIANCE BETWEEN INVENTORY AND TRANSFER SYSTEMATIC ERRORS
      MIN=MIN0(NSI,NST)
      JJ=MIN0(NMI,NMT)
      DO 310 I=1,NCI
      DO 310 I1=1,NCT
      DO 310 J=1,JJ
      KF=ITCI(I,I1,J)
      IF(KF.EQ.0)GO TO 310
      DO 300 K=1,KF
      CALL MTFIX(2,IB,I,J,K,XX1)
      CALL MTFIX(4,IT,I1,J,K,XX2)
      CVITS(I,I1,J,K)=XX1*XX2*SIS(I,J,K)*STS(I1,J,K)
  300 CONTINUE
  310 CONTINUE
C
C   COVARIANCE BETWEEN TRANSFER AT TIME K
C   AND INVENTORY AT TIME K+1
C
      MAX=MAX0(NSI,NST)
      DO 330 I=1,NCI
      DO 330 J=1,NMT
      DO 330 K=1,MAX
      IF(CVSIS(I,J,K) .LT. 1.E-9) GO TO 330
      DO 320 I1=1,NCT
      IF(CVITS(I,I1,J,K) .EQ.0) GO TO 320
      CALL MTFIX(2,IBP1,I,J,K,XX1)
      CALL MTFIX(4,IB,I1,J,K,XX2)
      XCVITS(I,I1,J,K)=XX1*XX2*SIS(I,J,K)*STS(I1,J,K)
  320 CONTINUE
  330 CONTINUE
  340 CONTINUE
C
C           COVARIANCE BETWEEN SUCCESSIVE TRANSFER SYSTEMATIC ERRORS
C
      DO 360 I=1,NCT
      DO 360 J=1,NMT
      KF=TSCI(I,J)
      IF(KF.EQ.0)GO TO 360
      DO 350 K=1,KF
      CALL MTFIX(4,ITP1,I,J,K,XX1)
      CALL MTFIX(4,IT,I,J,K,XX2)
      CVSTS(I,J,K)=XX1*XX2*VTS(I,J,K)
  350 CONTINUE
  360 CONTINUE
C
      DO 380 I=1,NCT
      DO 380 J=1,NMT
```

Fig. D-1. (cont)

```
      DO 380 K=1,NST
      IF(CVSTS(I,J,K) .LT. 1.E-09) GO TO 380
      DO 370 I1=1,NCT
      IF(CVTS(I,I1,J,K) .LT. 1.E-09) GO TO 370
      CALL MTFIX(4,ITP1,I1,J,K,XX1)
      CALL MTFIX(4,IT,I,J,K,XX2)
      IF(MOD(IT,NCAL(I,J,K)).EQ.0) XX1=0.
      XCVSTS(I,I1,J,K)=XX1*XX2*STS(I,J,K)*STS(I1,J,K)
  370 CONTINUE
  380 CONTINUE
C
      IF(MASPRT.EQ.0)GO TO 510
C
C *************** DEBUG OUTPUT
C
      CVITRX=0.
      WRITE(NPROUT,390)XINVT,VTIR,CVITRX
  390 FORMAT(///' TOTAL INVENTORY (XINVT) ='F12.6
     1 /' VARIANCE OF TOTAL RANDOM INVENTORY ERROR (VI) ='F12.6
     2 /' COVARIANCE BETWEEN INVENTORY-TRANSFER RANDOM ERRORS (CVITRX)='
     3 F12.6)
C
      WRITE(NPROUT,400)
  400 FORMAT(///' COVARIANCE OF INVENTORY BIAS ERROR - CVIS(NCI,NCI,NMI,
     1NSI)')
      CALL WRT4(CVIS,NCIMX,NCIMX,NMIMX,NCI,NCI,NMI,NSI,NPROUT)
C
      WRITE(NPROUT,410)
  410 FORMAT(///' COVARIANCE BETWEEN SUCCESSIVE INVENTORY SYSTEMATIC'
     1 ' ERRORS - CVSIS(NCI,NMI,NSI)')
      WRITE(NPROUT,420)
  420 FORMAT(/)
      CALL WRT3(CVSIS,NCIMX,NMIMX,NCI,NMI,NSI,NPROUT)
C
      WRITE(NPROUT,430)
  430 FORMAT(///' COVARIANCE BETWEEN SUCCESSIVE INVENTORY SYSTEMATIC'
     1 ' ERRORS FOR DIFFERENT INVENTORY COMPONENTS - XCVSIS(NCI,NCI,NMT'
     2 ',NST)')
      CALL WRT4(XCVSIS,NCIMX,NCIMX,NMIMX,NCI,NCI,NMI,NSI,NPROUT)
      IF(IB.EQ.1)GO TO 510
C
      WRITE(NPROUT,440)
  440 FORMAT(///' COVARIANCE BETWEEN INVENTORY-TRANSFER SYSTEMATIC'
     1 ' ERRORS - CVITB(NCI,NCT,NC,NSI)')
      CALL WRT4(CVITS,NCIMX,NCTMX,NMTMX,NCI,NCT,NC,NSI,NPROUT)
C
      WRITE(NPROUT,450)TR,VTTR
  450 FORMAT(///' TOTAL NET TRANSFER (TR) ='F10.6
     1  /' VARIANCE OF TOTAL RANDOM TRANSFER ERROR (VTTR) ='F10.6)
C
      WRITE(NPROUT,460)
  460 FORMAT(///' COVARIANCE OF TRANSFER SYSTEMATIC ERRORS'
     1 ' - CVTS(NCT,NCT,NMT,NST)')
      CALL WRT4(CVTS,NCTMX,NCTMX,NMTMX,NCT,NCT,NMT,NST,NPROUT)
C
      WRITE(NPROUT,470)
  470 FORMAT(///' COVARIANCE BETWEEN SUCESSIVE TRANSFERS - CVSTS(NCT,NMT
     1,NST)'/)
      WRITE(NPROUT,480)
  480 FORMAT(/)
      CALL WRT3(CVSTS,NCTMX,NMTMX,NCT,NMT,NST,NPROUT)
C
      WRITE(NPROUT,490)
  490 FORMAT(///' COVARIANCE BETWEEN SUCCESSIVE TRANSFER SYSTEMATIC'
     1 ' ERRORS FOR DIFFERENT TRANSFER COMPONENTS - XCVSTS(NCT,NCT,NMT'
```

Fig. D-1. (cont)

```
     2 ',NST)')
      CALL WRT4(XCVSTS,NCTMX,NCTMX,NMTMX,NCT,NCT,NMT,NST,NPROUT)
C
      WRITE(NPROUT,500)
  500 FORMAT(///' COVARIANCE BETWEEN SUCCESSIVE TRANSFER AND INVENTORY C
     1OMPONENTS - XCVITS(NCI,NCT,NMT,NST)')
      CALL WRT4(XCVITS,NCIMX,NCTMX,NMTMX,NCI,NCT,NMT,NST,NPROUT)
  510 CONTINUE
C
  520 FORMAT(8E15.8)
  530 FORMAT(20I4)
      RETURN
      END
      SUBROUTINE MASCUS
C
C ************** PROVIDES A BUFFER BETWEEN MASSAGE AND CUSUM.
C                 TAKES OUTPUT FROM MASSAGE AT EACH BALANCE TIME AND
C                 COMPUTES VARIANCES AND COVARIANCES REQUIRED FOR
C                 THE CUSUM VARIANCE CALCULATION.
C
C      VARIABLE DEFINITIONS:
C             S2I(I) - INVENTORY VARIANCE FOR THE I'TH BALANCE
C             S2H0   - INVENTORY VARIANCE AT THE INITIAL TIME
C             S2T(I)  - TRANSFER VARIANCE OVER THE ITH BALANCE INTERVAL
C             CVT(I)  - COVARIANCE BETWEEN ADJACENT TRANSFERS
C
       PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
C
C   PARAMETER DEFINITIONS:
C       NBALMX - MAXIMUM NUMBER OF BALANCES
C       NPVMX   - MAXIMUM NUMBER OF PV VARIABLES
C       NCIMX - MAXIMUM NUMBER OF INVENTORY COMPONENTS
C       NCTMX - MAXIMUM NUMBER OF TRANSFER COMPONENTS
C       NMIMX - MAXIMUM NUMBER OF INVENTORY MEASUREMENTS
C       NMTMX - MAXIMUM NUMBER OF TRANSFER MEASUREMENTS
C       NSIMX - MAXIMUM NUMBER OF INVENTORY SYSTEMATIC ERRORS
C       NSTMX - MAXIMUM NUMBER OF TRANSFER SYSTEMATIC ERRORS
C       NCMX  - MINIMUM OF NMIMX AND NMTMX
C       MXPMX - MAXIMUM OF NSIMX AND NSTMX
C       NSTRMX - MAXIMUM NUMBER OF RANDOM NUMBER STREAMS
C       NCOLMX - MAXIMUM NUMBER OF PULSE COLUMNS
C
       COMMON /CVCOM/ CVSIS(NCIMX,NMIMX,NSIMX), CVITS(NCIMX,NCTMX,NMTMX,
     + MXPMX), CVTS(NCTMX,NCTMX,NMTMX,NSTMX), CVSTS(NCTMX,NMTMX,NSTMX),
     + CVIS(NCIMX,NCIMX,NMIMX,NSIMX), CVITR(NCIMX,NCTMX,NMTMX),VTIR,
     + XCVSIS(NCIMX,NCIMX,NMIMX,NSIMX), XCVSTS(NCTMX,NCTMX,NMTMX,NSTMX),
     + VTTR, XCVITS(NCIMX,NCTMX,NMTMX,MXPMX)
C
       COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
       COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
C
C
       INTEGER TSCI,TCI
```

Fig. D-1. (cont)

```
C
      COMMON /MSINCM/ ICI(NCIMX,NCIMX,NMIMX), ISCI(NCIMX,NMIMX),
     + ITCI(NCIMX,NCTMX,NCMX), TCI(NCTMX,NCTMX,NMTMX),TSCI(NCTMX,NMTMX),
     +  VIR(NCIMX,NMIMX), VTR(NCTMX,NMTMX),VIS(NCIMX,NMIMX,2),
     +  VTS(NCTMX,NMTMX,2), SIR(NCIMX,NMIMX),STR(NCTMX,NMTMX),
     +  SIS(NCIMX,NMIMX,2), STS(NCTMX,NMTMX,2)
C
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DIMENSION SRT(NCTMX,NMTMX), SST(NCTMX,NMTMX,2)
C
C ******** HOLDUP AND TRANSFER VARIANCE AND COVARIANCE CALCULATIONS
C
C     TRACE 300
C
C *************** ZERO OUT VARIANCE AND COVARIANCE ARRAYS
C
      DO 10 I=1,NCI
      DO 10 K=1,NMI
      DO 10 L=1,NSI
      CVSIS(I,K,L)=0.
      DO 10 J=1,NCI
      CVIS(I,J,K,L)=0.
      XCVSIS(I,J,K,L)=0.
   10 CONTINUE
      DO 20 I=1,NCT
      DO 20 K=1,NMT
      DO 20 L=1,NST
      CVSTS(I,K,L)=0.
      DO 20 J=1,NCT
      CVTS(I,J,K,L)=0.
      XCVSTS(I,J,K,L)=0.
   20 CONTINUE
      DO 30 I=1,NCI
      DO 30 J=1,NCT
      DO 30 K=1,NMT
      CVITR(I,J,K)=0.
      DO 30 L=1,MXP
      CVITS(I,J,K,L)=0.
      XCVITS(I,J,K,L)=0.
   30 CONTINUE
      IT=0
      DO 130 IB=1,NBALP1
      DO 130 III=1,NTRPBL
      IT=IT+1
      CALL MASAGE(IB,IT,III)
      IF(III.NE.1)GO TO 60
C
C ************** VARIANCE OF TOTAL INVENTORY RANDOM ERROR
C
      S2I(IB)=VTIR
      S2IR(IB)=VTIR
C
C ************** VARIANCE OF INVENTORY SYSTEMATIC ERRORS
C
      DO 40 I=1,NCI
      DO 40 J=1,NCI
      DO 40 K=1,NMI
```

Fig. D-1. (cont)

```
      DO 40 L=1,NSI
      S2I(IB)=S2I(IB)+CVIS(I,J,K,L)
   40 CONTINUE
      S2IB(IB)=S2I(IB)-VTIR
C
C ************** COVARIANCE BETWEEN SUCCESSIVE INVENTORIES
C
      CVI(IB)=0.
      DO 50 I=1,NCI
      DO 50 J=1,NMI
      DO 50 K=1,NSI
      CVI(IB)=CVI(IB)+CVSIS(I,J,K)
   50 CONTINUE
C
C ************** VARIANCE OF TOTAL TRANSFER RANDOM ERROR
C
   60 S2T(IT)=VTTR
      S2TR(IT)=VTTR
C
C *************** VARIANCE OF TRANSFER SYSTEMATIC ERRORS
C
      DO 70 I=1,NCT
      DO 70 J=1,NCT
      DO 70 K=1,NMT
      DO 70 L=1,NST
      S2T(IT)=S2T(IT)+CVTS(I,J,K,L)
   70 CONTINUE
      S2TB(IT)=S2T(IT)-VTTR
C
C *************** COVARIANCE BETWEEN SUCCESSIVE TRANSFERS
C
      CVT(IT)=0.
      DO 80 I=1,NCT
      DO 80 I1=1,NCT
      DO 80 J=1,NMT
      DO 80 K=1,NST
      CVT(IT)=CVT(IT)+XCVSTS(I,I1,J,K)
   80 CONTINUE
C
C *************** OUTPUT TO DECANAL FOR THE CASE OF COLLAPSED INVENTORY
C *************** AND NON-COLLAPSED TRANSFERS
C
      IF(ICLAPS.NE.1)GO TO 130
      IF(III.EQ.1)WRITE(NDECIN,110)XIT(IB)
      WRITE(NDECIN,110) ((TM(IB,I,J),J=1,NMT),I=1,NCT)
      WRITE(NDECIN,120)IT
      IF(IB.NE.1)GO TO 130
      IF(III.NE.1)GO TO 130
      SRI=SQRT(S2IR(1))/XIT(1)
      SSI=SQRT(S2IB(1))/XIT(1)
      DO 90 I=1,NCT
      DO 90 J=1,NMT
      SRT(I,J)=SQRT(VTR(I,J))
      DO 90 K=1,NST
      SST(I,J,K)=SQRT(VTS(I,J,K))
   90 CONTINUE
      WRITE(NDECIN,110) SRI
      WRITE(NDECIN,110)((SRT(I,J),J=1,NMT),I=1,NCT)
      WRITE(NDECIN,110) SSI
      WRITE(NDECIN,110)(((SST(I,J,K),K=1,NST),J=1,NMT),I=1,NCT)
      ICI(1,1,1)=1
      ISCI(1,1)=1
      DO 100 I=1,NCT
      ITCI(1,I,1)=0
  100 CONTINUE
```

Fig. D-1. (cont)

```
      WRITE(NDECIN,12Ø) ICI(1,1,1)
      WRITE(NDECIN,12Ø)(((TCI(1,I1,J),J=1,NMT),I1=1,NCT),I=1,NCT)
      WRITE(NDECIN,12Ø)(ITCI(1,I,1),I=1,NCT)
      WRITE(NDECIN,12Ø) ISCI(1,1)
      WRITE(NDECIN,12Ø)((TSCI(I,J),J=1,NMT),I=1,NCT)
  11Ø FORMAT(8G15.7)
  12Ø FORMAT(2ØI4)
  13Ø CONTINUE
      RETURN
      END
      SUBROUTINE MEASR(XPROC,XMES,N,IPV)
C
C ********************************************************************
C       THIS ROUTINE ADDS MEASUREMENT ERRORS TO THE ACTUAL VALUES
C ********************************************************************
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
      COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
C
      COMMON/ MSRNRM/ RN1,RN2,ISTRM
C
      DIMENSION XPROC(1),XMES(1),ETA(2),ETASAV(2),ICALBC(2),K(2), KK(2)
C
C       ************ VARIABLES *************
C
C          XPROC  - ARRAY OF PROCESS VALUES
C          XMES   - ARRAY OF MEASURED VALUES
C          N      - NUMBER OF MEASUREMENTS TO BE COMPUTED
C          IPV    - PROCESS VARIABLE NUMBER
C          EPS    - RANDOM ERROR
C          ETA    - SYSTEMATIC ERROR
C          ICALBC - CALIBRATION COUNTER
C          IZE    - ZERO ERROR FLAG
C                         .EQ. Ø - NONZERO MEASUREMENTS ERRORS
C                         .EQ. 1 - ZERO MEASUREMENT ERRORS
C          INTCAL - NUMBER OF BATCHES BETWEEN CALIBRATIONS
C          ISTRPV - RANDOM NUMBER STREAM FOR PROCESS VARIABLE RANDOM
C                   ERROR COMPUTATION  (STREAM NUMBERS FOR SYSTEMATIC
C                   ERRORS ARE ISTRPV + 1 AND 2.)
C          MESTYP - MEASUREMENT TYPE CORRESPONDING TO A GIVEN PROCESS
C                   VARIABLE.
C                      .EQ. 1 - MULTIPLICATIVE MODEL
C                      .EQ. 2 - ADDITIVE MODEL
C                      .EQ. 3 - HYBRID MODEL (MULTIPLICATIVE FOR RANDOM
C                               AND SHORT TERM SYSTEMATIC ERRORS:
C                               ADDITIVE FOR LONG-TERM SYSTEMATIC ERRORS)
C          SIGMAE - RANDOM ERROR STANDARD DEVIATION
C          SIGMAN - SYSTEMATIC ERROR STANDARD DEVIATION
C
C ****************
C
C ********************************************************************
C               COMPUTE FIRST TWO ETA VALUES (SYSTEMATIC ERRORS)
C ********************************************************************
C
C
```

Fig. D-1. (cont)

```
C     TRACE 110
      ISTRMR=ISTRNR(IPV)
      CALL DRAND(-ISTRMR)
      DO 20 JK=1,2
      ICALBC(JK)=0
      IF(IPVTRN(IPV).GT.0)ICALBC(JK)=-1
      IF(SIGMAN(IPV,JK).NE.0) GO TO 10
      ETA(JK)=0.0E0
      ETASAV(JK)=0.0E0
      GO TO 20
   10 CONTINUE
      ISTRM=ISTRNS(IPV)
      IF(JK.EQ.2)ISTRM=ISTRM+1
      CALL DRAND(-ISTRM)
      CALL RNORM
      K(JK)=1
      KK(JK)=1
      ETA(JK)=RN1*SIGMAN(IPV,JK)
      ETASAV(JK)=RN2*SIGMAN(IPV,JK)
   20 CONTINUE
      J=2
      JJ=0
   30 DO 140 I=1,N
      DO 60 JK=1,2
      IF(SIGMAN(IPV,JK).EQ.0) GO TO 60
      ICALBC(JK)=ICALBC(JK)+1
C
C ***********************************************************************
C                CHECK FOR RECALIBRATION
C ***********************************************************************
C
      IF(ICALBC(JK).LE.INTCAL(IPV,JK)) GO TO 60
      ICALBC(JK)=1
      K(JK)=K(JK)+1
      IF(K(JK).EQ.3) GO TO 40
      ETA(JK)=ETASAV(JK)
      GO TO 60
   40 ISTRM=ISTRMR+JK
      CALL RNORM
      K(JK)=1
      KK(JK)=KK(JK)+1
      IF(KK(JK).EQ.2) GO TO 50
      ETA(JK)=RN1*SIGMAN(IPV,JK)
      ETASAV(JK)=RN2*SIGMAN(IPV,JK)
      GO TO 60
C
C ***********************************************************************
C                FLIP-FLOP RANDOM NUMBERS
C ***********************************************************************
C
   50 ETASAV(JK)=RN1*SIGMAN(IPV,JK)
      ETA(JK)=RN2*SIGMAN(IPV,JK)
      KK(JK)=0
   60 CONTINUE
      J=J+1
      ISTRM=ISTRMR
      IF(J.EQ.3) GO TO 70
      EPS=EPSSAV
      GO TO 90
   70 CALL RNORM
      J=1
      JJ=JJ+1
      IF(JJ.EQ.2) GO TO 80
      EPS=RN1*SIGMAE(IPV)
      EPSSAV=RN2*SIGMAE(IPV)
```

Fig. D-1. (cont)

```
      GO TO 90
C
C ******************************************************************
C                  FLIP-FLOP RANDOM NUMBERS
C ******************************************************************
C
   80 EPSSAV=RN1*SIGMAE(IPV)
      EPS=RN2*SIGMAE(IPV)
      JJ=0
   90 CONTINUE
C
C ******************************************************************
C                 CHOOSE THE CORRECT MEASUREMENT TYPE
C ******************************************************************
C
      MESM2=MESTYP(IPV)-2
      IF(MESM2.LT.-1)GO TO 100
      IF(MESM2.GT.1)GO TO 100
      IF(MESM2)110,120,130
C
C ******************************************************************
C                 ZERO ERROR MODEL
C ******************************************************************
C
  100 CONTINUE
      XMES(I)=XPROC(I)
      GO TO 140
C
C ******************************************************************
C                 MULTIPLICATIVE MODEL
C ******************************************************************
C
  110 XMES(I)=XPROC(I)*(1.0E0+EPS+ETA(1)+ETA(2))
      GO TO 140
C
C ******************************************************************
C                 ADDITIVE MODEL
C ******************************************************************
C
  120 XMES(I)=XPROC(I)+EPS+ETA(1)+ETA(2)
      GO TO 140
C
C *********************************************************************
C                 HYBRID MODEL
C                     MULTIPLICATIVE FOR RANDOM AND SHORT-TERM SYSTEMATIC ERRORS:
C                     ADDITIVE FOR LONG-TERM SYSTEMATIC ERRORS
C*********************************************************************
C
  130 XMES(I)=XPROC(I)*(1.+EPS+ETA(1)) + ETA(2)
C
  140 CONTINUE
      RETURN
      END
      SUBROUTINE MESDRV
C
C ******************************************************************
C               CONTROLLER FOR PROCESS MEASUREMENT ROUTINES
C ******************************************************************
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
```

Fig. D-1. (cont)

```
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /PVCOMS/ PVIS(NBMXP3,NPVIMX),PVTS(NTMXP2,NPVTMX)
      COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
      COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
      COMMON /WORK/ X(NTMXP2,3)
C
C
C ***************** ZERO OUT ARRAYS *****************
C
C     TRACE 1000
      ICINV=0
      ICTRN=0
      CALL ZERO
C
C ************** CONVERT PROCESS VALUES TO MEASURED VALUES
C
      IF(IZE.EQ.0)GO TO 50
      DO 40 II=1,NPVCNT
      I=IPVNAR(II)
      IPVN=NPVIT(I)
      IF(IPVTRN(I).GT.0)GO TO 20
      DO 10 J=1,NBALP2
      PVI(J,IPVN)=PVIS(J,IPVN)
   10 CONTINUE
      GO TO 40
   20 DO 30 J=1,NTRNP2
      PVT(J,IPVN)=PVTS(J,IPVN)
   30 CONTINUE
   40 CONTINUE
      GO TO 110
   50 CONTINUE
      DO 100 II=1,NPVCNT
      I=IPVNAR(II)
      IPVN=NPVIT(I)
      IF(IPVTRN(I))60,80,90
   60 DO 70 J=1,NBALP2
      PVI(J,IPVN)=PVIS(J,IPVN)
   70 CONTINUE
      GO TO 100
   80 CONTINUE
      CALL MEASR(PVIS(1,IPVN),PVI(1,IPVN),NBALP2,I)
      GO TO 100
   90 CALL MEASR(PVTS(1,IPVN),PVT(1,IPVN),NTRNP2,I)
  100 CONTINUE
  110 IF(ISPNTI.EQ.0)GO TO 120
C
C ************** ISOLATED UNIT PROCESS COMPUTATIONS
C
      IF(ITIPRP.EQ.1)WRITE(1,130)ISPNTI,(IPVNO(I,ISPNTI),I=1,3)
      CALL PROCES(IPVNO(1,ISPNTI),ISPNTI)
      GO TO 150
```

Fig. D-1. (cont)

```
  120 CONTINUE
C
C *************** PROCESS CALCULATIONS!
C
      DO 140 ITRIN=1,NTRIN
      IF(ITIPRP.EQ.1)WRITE(1,130)ITRIN,(IPVNO(I,ITRIN),I=1,3)
  130 FORMAT(' PROCESS NUMBER ='I3'   PV VARIABLE NUMBER ='I3)
      CALL PROCES(IPVNO(1,ITRIN),ITRIN)
  140 CONTINUE
  150 CALL PRTBAL
      IF(NRUN.NE.1)RETURN
      IF(ICLAPS.EQ.2)CALL OUTDEC
  160 RETURN
      END
      SUBROUTINE MESIN
C
C ***********************************************************************
C                         READS INPUT DATA
C ***********************************************************************
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
C   PARAMETER DEFINITIONS:
C       NBALMX - MAXIMUM NUMBER OF BALANCES
C       NPVMX    - MAXIMUM NUMBER OF PV VARIABLES
C       NCIMX  - MAXIMUM NUMBER OF INVENTORY COMPONENTS
C       NCTMX  - MAXIMUM NUMBER OF TRANSFER COMPONENTS
C       NMIMX  - MAXIMUM NUMBER OF INVENTORY MEASUREMENTS
C       NMTMX  - MAXIMUM NUMBER OF TRANSFER MEASUREMENTS
C       NSIMX  - MAXIMUM NUMBER OF INVENTORY SYSTEMATIC ERRORS
C       NSTMX  - MAXIMUM NUMBER OF TRANSFER SYSTEMATIC ERRORS
C       NCMX   - MINIMUM OF NMIMX AND NMTMX
C       MXPMX  - MAXIMUM OF NSIMX AND NSTMX
C       NSTRMX - MAXIMUM NUMBER OF RANDOM NUMBER STREAMS
C       NCOLMX - MAXIMUM NUMBER OF PULSE COLUMNS
C
C
      INTEGER VRTITL(5,5)
      INTEGER*4 ISEED, LSEED
      COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /PVCOMS/ PVIS(NBMXP3,NPVIMX),PVTS(NTMXP2,NPVTMX)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /SEED/ ISEED(NSTRMX), LSEED(NSTRMX)
C
C
      COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
C
```

Fig. D-1. (cont)

```
      INTEGER*2 DCNLIN, PVARA
      COMMON /TITLE/ ITITLE(40),IPVTI(30,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
      DIMENSION IDATE(8),ITIME(4)
C
C
      DATA VRTITL/10HVOLUME    =,10HPU CONC. =,10HKG PU     =,
     1        10HPU HOLDUP=,10HFLOW RATE=/
C
C
C                 ******** VARIABLES *********
C
C           IZE     - ZERO ERROR FLAG
C                     .EQ. 0 - MEASUREMENT ERRORS INCLUDED
C                     .EQ. 1 - MEASUREMENT ERRORS NOT INCLUDED
C           NRUN    - NUMBER OF RUNS
C           NPV   - NUMBER OF PV VARIABLES
C           NINV  - NUMBER OF INVENTORIES
C           ISPNTI  - SPECIFIC PROCESS NUMBER (ISOLATION RUN)
C           NBAL    - NUMBER OF MATERIAL BALANCES
C
C
      CALL DATE$A(IDATE)
      CALL TIME$A(ITIME)
      WRITE(NPROUT,10) (IDATE(I), I=1,8), (ITIME(I), I=1,4)
   10 FORMAT(/1X,8A2,5X,4A2)
C
C **************** READ AND PRINT TITLE FROM PV ARRAY FILE
C
      READ(NPVIN,20) (ITITLE(I),I=1,40)
   20 FORMAT(40A2)
      WRITE(1,30) (ITITLE(I),I=1,40)
      WRITE(NPROUT,40) (ITITLE(I),I=1,40)
   30 FORMAT(' *****************************************************'
     1 //3X,40A2//
     2 ' *****************************************************')
   40 FORMAT(/,1X,40A2)
C
C ****************
C
C ********************************************************************
C                 READ INPUT DATA FROM MESDAT FILE
C ********************************************************************
C
      CALL BLANKS(6)
      READ(NDAT,50) (ITITLE(I), I=1,40)
   50 FORMAT(40A2)
      WRITE(NPROUT,60)(ITITLE(I),I=1,40)
   60 FORMAT(/1X,40A2)
      IZE=0
      IRNSCH=0
      NRUN=1
      NTRIN=1
      ISPNTI=0
      NBAL=1
      NTRPBL=1
      DT=1.
      MASPRT=0
      ITIPRP=0
      IMESPR=0
      ICLAPS=0
      NPV=1
      READ(NDAT,*) IBLANK
      READ(NDAT,*) IZE
```

Fig. D-1. (cont)

```
      READ(NDAT,*) IRNSCH
      READ(NDAT,*) NRUN
      READ(NDAT,*) NBAL
      READ(NDAT,*)NTRPBL
      READ(NDAT,*)DT
      READ(NDAT,*)MASPRT
      READ(NDAT,*)ITIPRP
      READ(NDAT,*)IMESPR
      READ(NDAT,*)IPVPRT
      READ(NDAT,*)ICLAPS
      WRITE(NPROUT,70) IZE,IRNSCH,NRUN,NBAL,NTRPBL,DT,MASPRT,ITIPRP,
     + IMESPR,IPVPRT,ICLAPS
   70 FORMAT(/5X'ZERO ERROR FLAG (1 GIVES ZERO ERROR CASE) (IZE) ='I2
     $ /5X'FLAG FOR CHANGING RANDOM NUMBER SEEDS (IRNSCH) ='I2
     $ /5X'NUMBER/OF RUNS (NRUN) ='I4
     2 /5X'NUMBER/OF BALANCES (NBAL) ='I3
     $ /5X'NUMBER/OF TRANSFERS PER BALANCE (NTRPBL) ='I3
     3 /5X'TIME INTERVAL (DT) ='F8.3
     5 /5X'MASSAGE DEBUGG PRINT FLAG (MASPRT) ='I2
     6 /5X'TRANSFER-INVENTORY AND PROCESS VARIABLE NO. PRINT FLAG (ITIPR
     $P) ='I2
     7 /5X'PRINTOUT FLAG FOR INPUT MEASUREMENT ERRORS (IMESPR) ='I2
     8 /5X'PRINTOUT FLAG FOR INPUT PROCESS VARIABLES (IPVPRT) ='I2
     9 /5X'ICLAPS/(COLLAPSE MATRIX OUTPUT TO SCALARS WHEN .GT. 0) ='I2)
C
C *************** READ IN PROCESS DEPENDENT DATA
C
C
C ***************************************************************************
C                        INPUT PROCESS CONTROL PARAMETERS
C ***************************************************************************
C
      CALL BLANKS(3)
      WRITE(NPROUT,80)
   80 FORMAT(//' ****************************************************'/
     1 ' INPUT DATA FOR DEFINING UNIT PROCESS ACCOUNTING AREA (UPAA)')
      READ(NDAT,*)NPV
      WRITE(NPROUT,90) NPV
   90 FORMAT(/5X'NUMBER OF PROCESS VARIABLES (NPV) ='I3)
      READ(NDAT,*)NTRIN
      WRITE(NPROUT,100)NTRIN
  100 FORMAT(5X'NUMBER OF TRANSFER-INVENTORIES (NTRIN) ='I3)
      DO 110 I=1,NTRIN
      ITIN(I)=1
      DO 110 J=1,5
      IPVNO(J,I)=0
  110 CONTINUE
      CALL BLANKS(2)
      READ(NDAT,*)(ITIN(I),I=1,NTRIN)
      WRITE(NPROUT,120)(ITIN(I),I=1,NTRIN)
  120 FORMAT(5X'ARRAY OF TRANSFER-INVENTORY NUMBERS (ITIN)'/(5X,40I3))
C
C *************** SET TRANSFER DIRECTION INDICATORS
C
      DO 140 I=1,NTRIN
      ITRAN(I)=0
      IF(ITIN(I).LT.0)GO TO 130
      IF(ITIN(I).LT.5)GO TO 140
      ITRAN(I)=1
      GO TO 140
  130 ITIN(I)=-ITIN(I)
      ITRAN(I)=-1
  140 CONTINUE
C
C *************** SET PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH
```

Fig. D-1. (cont)

```
C *************** TRANSFER-INVENTORY NUMBER
C
      CALL BLANKS(2)
      WRITE(NPROUT,150)
  150 FORMAT(/5X'ARRAY OF PROCESS VARIABLE NUMBERS ASSOCIATED WITH EACH
     1TRANSFER OR INVENTORY (IPVNO)'//10X'TRANSFER'/10X'INVENTORY'12X
     2 'PROCESS VARIABLE NUMBER'/11X'NUMBER'8X'(1)'5X'(2)'5X'(3)'5X'(4)'
     3 5X'(5)')
      DO 170 J=1,NTRIN
      READ(NDAT,*) (IPVNO(I,J),I=1,5)
      WRITE(NPROUT,160) J, (IPVNO(I,J),I=1,5)
  160 FORMAT(13X,I2,9X,I3,5X,I3,5X,I3,5X,I3,5X,I3)
  170 CONTINUE
      CALL BLANKS(1)
      READ(NDAT,*) ISPNTI
      WRITE(NPROUT,180) ISPNTI
  180 FORMAT(/5X'SPECIFIC TRANSFER-INVENTORY NUMBER (ISPNTI) ='I3)
C
C *************** READ PV ARRAY
C
      WRITE(NPROUT,190)
  190 FORMAT(/' *********************************************************'
     1 /' BEGIN READING IN PROCESS VARIABLE ARRAY')
      CALL REDPV
      WRITE(NPROUT,200)
  200 FORMAT(/' READING OF PROCESS VARIABLE ARRAY COMPLETE'
     1 /' *********************************************************')
C
C **********************************************************************
C                   INITIALIZE STATISTICAL PARAMETERS
C **********************************************************************
C
      CALL BLANKS(3)
      WRITE(NPROUT,210)
  210 FORMAT(/' MEASUREMENT ERRORS FOR EACH PROCESS VARIABLE')
      DO 220 I=1,NPV
      SIGMAE(I)=0.0
      MESTYP(I)=1
      DO 220 J=1,2
      SIGMAN(I,J)=0.0
  220 INTCAL(I,J)=10000
      DO 230 I=1,NPV
      CALL READEM(I)
      SIG2E(I)=SIGMAE(I)**2
      DO 230 J=1,2
      SIG2N(I,J)=SIGMAN(I,J)**2
  230 CONTINUE
      KI=0
      KT=0
      DO 260 IPV=1,NPV
      WRITE(NPROUT,240)(IPVTI(I,IPV),I=1,30)
  240 FORMAT(/5X,30A2/)
      IF(IPVTRN(IPV).GT.0)GO TO 250
      KI=KI+1
      WRITE(NPROUT,270) PVIS(1,KI)
      GO TO 260
  250 KT=KT+1
      WRITE(NPROUT,270) PVTS(1,KT)
  260 CALL WRITEM(IPV)
  270 FORMAT(9X,'INITIAL VALUE ='F12.6)
C
C *************** ECHO CHECK PROCESS VARIABLES
C
      IF(IPVPRT.EQ.0)GO TO 330
      WRITE(NPROUT,280)
```

Fig. D-1. (cont)

```
  280 FORMAT(///1H1,'ECHO OUTPUT PROCESS VARIABLE FILE')
      KI=0
      KT=0
      DO 320 IPV=1,NPV
      WRITE(NPROUT,290)(IPVTI(I,IPV),I=1,30)
  290 FORMAT(///,1X,30A2)
      NFC=NPVCT(IPV)
      IF(IPVTRN(IPV).GT.0)GO TO 310
      KI=KI+1
      WRITE(NPROUT,300)(PVIS(I,KI),I=1,NFC)
  300 FORMAT(10E12.4)
      GO TO 320
  310 KT=KT+1
      WRITE(NPROUT,300)(PVTS(I,KT),I=1,NFC)
  320 CONTINUE
  330 CONTINUE
      WRITE(NPROUT,340)
  340 FORMAT(/' ***********************************************************'
     1 //' IMPORTANT INTEGERS CALCULATED IN SUBROUTINE SETMAS')
      CALL SETMAS
      WRITE(NPROUT,350) NPVCNT,NPVI,NPVT,NCI,NCT,NMI,NMT,NSI,NST,NNSTRM,
     + NCOLUM
  350 FORMAT(/5X'NUMBER OF PROCESS VARIABLES USED FOR THIS CASE (NPVCNT)
     +  ='I3/5X'NUMBER OF INVENTORY PROCESS VARIABLES (NPVI)
     +  ='I3/5X'NUMBER OF TRANSFER PROCESS VARIABLES (NPVT)
     +  ='I3/5X'NUMBER OF INVENTORY COMPONENTS (NCI)
     +  ='I3/5X'NUMBER OF TRANSFER COMPONENTS (NCT)
     +  ='I3/5X'NUMBER OF INVENTORY MEASUREMENTS (NMI)
     +  ='I3/5X'NUMBER OF TRANSFER MEASUREMENTS (NMT)
     +  ='I3/5X'NUMBER OF INVENTORY SYSTEMATIC ERRORS (NSI)
     +  ='I3/5X'NUMBER OF TRANSFER SYSTEMATIC ERRORS (NST)
     +  ='I3/5X'NUMBER OF RANDOM NUMBER STREAMS (NNSTRM)
     +  ='I3/5X'NUMBER OF PULSE COLUMNS (NCOLUM) ='I3)
      WRITE(NPROUT,360)
  360 FORMAT(/' ***********************************************************')
C
C *************** CHECK IF ANY CONTROL INTEGERS WILL CAUSE ARRAY OVERFLOW
C
      IF(NPV.LE.NPVMX)GO TO 380
      WRITE(NPROUT,370)
      WRITE(1,370)
  370 FORMAT(' RUN TERMINATED WITH NPV GREATER THAN NPVMX!)
      CALL CLOSEM
  380 IF(NCI.LE.NCIMX)GO TO 400
      WRITE(NPROUT,390)
      WRITE(1,390)
  390 FORMAT(' RUN TERMINATED WITH NCI GREATER THAN NCIMX!)
      CALL CLOSEM
  400 IF(NCT.LE.NCTMX)GO TO 420
      WRITE(NPROUT,410)
      WRITE(1,410)
  410 FORMAT(' RUN TERMINATED WITH NCT GREATER THAN NCTMX!)
      CALL CLOSEM
  420 IF(NMI.LE.NMIMX)GO TO 440
      WRITE(NPROUT,430)
      WRITE(1,430)
  430 FORMAT(' RUN TERMINATED WITH NMI GREATER THAN NMIMX')
      CALL CLOSEM
  440 IF(NMT.LE.NMTMX)GO TO 460
      WRITE(NPROUT,450)
      WRITE(1,450)
  450 FORMAT(' RUN TERMINATED WITH NMT GREATER THAN NMTMX!)
      CALL CLOSEM
  460 IF(NSI.LE.NSIMX)GO TO 480
      WRITE(NPROUT,470)
```

Fig. D-1. (cont)

```
      WRITE(1,47Ø)
  47Ø FORMAT(' RUN TERMINATED WITH NSI GREATER THAN NSIMX')
      CALL CLOSEM
  48Ø IF(NST.LE.NSTMX)GO TO 5ØØ
      WRITE(NPROUT,49Ø)
      WRITE(1,49Ø)
  49Ø FORMAT(' RUN TERMINATED WITH NST GREATER THAN NSTMX')
      CALL CLOSEM
  5ØØ IF(NNSTRM.LE.NSTRMX)GO TO 52Ø
      WRITE(NPROUT,51Ø)
      WRITE(1,51Ø)
  51Ø FORMAT(' RUN TERMINATED WITH NNSTRM GREATER THAN NSTRMX')
      CALL CLOSEM
  52Ø IF(NCOLUM.LE.NCOLMX)GO TO 54Ø
      WRITE(NPROUT,53Ø)
      WRITE(1,53Ø)
  53Ø FORMAT(' RUN TERMINATED WITH NCOLUM GREATER THAN NCOLMX')
      CALL CLOSEM
  54Ø NC=MINØ(NMI,NMT)
      MXP=MAXØ(NSI,NST)
      IF(NC.LE.NCMX)GO TO 56Ø
      WRITE(NPROUT,55Ø)
      WRITE(1,55Ø)
  55Ø FORMAT(' RUN TERMINATED WITH NC GREATER THAN NCMX')
      CALL CLOSEM
  56Ø IF(MXP.LE.MXPMX)GO TO 58Ø
      WRITE(NPROUT,57Ø)
      WRITE(1,57Ø)
  57Ø FORMAT(' RUN TERMINATED WITH MXP GREATER THAN MXPMX')
      CALL CLOSEM
  58Ø IF(NBAL.LE.NBALMX)GO TO 6ØØ
      WRITE(NPROUT,59Ø)
      WRITE(1,59Ø)
  59Ø FORMAT(' RUN TERMINATED WITH NBAL GREATER THAN NBALMX')
      CALL CLOSEM
  6ØØ CONTINUE
      IF(NCOLUM.EQ.Ø)GO TO 65Ø
C
C ************ READ INPUT DATA ASSOCIATED WITH THE COLUMNS
C
      CALL BLANKS(4)
      WRITE(NPROUT,61Ø)
  61Ø FORMAT(/' COLUMN CONSTANTS (HCC(5,I), VCC(3,I), CCC(3,I)')
      DO 62Ø J=1,NCOLUM
      READ(NDAT,*)(HCC(I,J),I=1,5), (VCC(I,J),I=1,2), (CCC(I,J),I=1,3)
      WRITE(NPROUT,63Ø)(HCC(I,J),I=1,5), (VCC(I,J),I=1,2),(CCC(I,J),I=1,
     +  3)
  62Ø CONTINUE
  63Ø FORMAT(8E14.6)
C
C ************ SET UP ARRAY OF COLUMN NUMBERS
C
      IC=Ø
      DO 64Ø I=1,NTRIN
      NCOL(I)=Ø
      IF(ITIN(I).LT.3)GO TO 64Ø
      IF(ITIN(I).GT.4)GO TO 64Ø
      IC=IC+1
      NCOL(I)=IC
  64Ø CONTINUE
C
C *******************************************************************
C                         INITIAL CALCULATIONS
C *******************************************************************
C
```

Fig. D-1. (cont)

```
C
C *********** MINIMUM NUMBER OF TRANSFER AND INVENTORY ENTRIES PER P. V.
C
  650 NMININ=10000
      NMINTR=10000
      DO 670 I=1,NPV
      IF(IPVTRN(I).GT.0)GO TO 660
      IF(NMININ.GT.NPVCT(I))NMININ=NPVCT(I)
      GO TO 670
  660 IF(NMINTR.GT.NPVCT(I))NMINTR=NPVCT(I)
  670 CONTINUE
  680 NBALP1=NBAL+1
      NBALP2=NBAL+2
      NCITST=NCI
      IF(NCOLUM.GT.0)NCITST=NCI-1
      NTRN=NTRPBL*NBALP1+1
      NTRNP1=NTRN+1
      NTRNP2=NTRN+2
C
C ************ CHECK FOR MINIMUM NUMBER OF INVENTORY AND TRANSFER ENTRIES
C ************ PER PROCESS VARIABLE
C
      IF(NBALP2.LE.NMININ)GO TO 700
      WRITE(NPROUT,690) NMININ, NBALP2
      WRITE(1,690) NMININ, NBALP2
  690 FORMAT(//' YOU HAVE ONLY 'I3' INVENTORY PROCESS VARIABLE ENTRIES P
     1ER PROCESS VARIABLE'/' A MINIMUM OF 'I3' (NBAL+2) ARE REQUIRED'/
     2 /' ******* RUN IS TERMINATED *******')
      CALL CLOSEM
  700 IF(NTRNP2.LE.NMINTR) GO TO 720
      WRITE(NPROUT,710) NMINTR,NTRNP2
      WRITE(1,710) NMINTR,NTRNP2
  710 FORMAT(//' YOU HAVE ONLY 'I3' TRANSFER PROCESS VARIABLE ENTRIES PE
     1R PROCESS VARIABLE'/' A MINIMUM OF 'I3' ((NBAL+1)*NTRPBL+3) ARE RE
     2QUIRED'/ ' ******* RUN TERMINATED *******')
      CALL CLOSEM
  720 CONTINUE
C
C ************** CHECK FOR OVERFLOW OF TRANSFER ARRAYS
C
      IF(NTRN.LE.NTRNMX)GO TO 740
      WRITE(NPROUT,730)
      WRITE(1,730)
  730 FORMAT('RUN TERMINATED WITH NTRN GREATER THAN NTRNMX')
      CALL CLOSEM
  740 CONTINUE
C
C ************** INITIALIZE INVENTORY AND TRANSFER ELEMENTS
C
      DO 750 I=1,NBALP2
      DO 750 J=1,NCI
      DO 750 K=1,NMI
      XIM(I,J,K)=0.
  750 CONTINUE
      DO 760 I=1,NTRNP2
      DO 760 J=1,NCT
      DO 760 K=1,NMT
      TM(I,J,K)=0.
  760 CONTINUE
C
C ***********************************************************************
C                         INPUT INITIAL RANDOM NUMBER SEEDS
C ***********************************************************************
C
      DO 770 I=1,NNSTRM
```

Fig. D-1. (cont)

```
      READ(NFSEED,*)ISEED(I)
      LSEED(I)=ISEED(I)
  770 CONTINUE
      WRITE(NPROUT,780)(ISEED(I),I=1,NNSTRM)
  780 FORMAT(/' INITIAL RANDOM NUMBER SEEDS'/(10I12))
C
C ********* SET THE RANDOM NUMBER STREAM ASSOCIATED WITH
C ********* EACH PROCESS VARIABLE
C
      RETURN
      END
      SUBROUTINE MTFIX(MT,IB,NC,NM,NS,XX)
C
C ************ COMPENSATES FOR MEASUREMENT ERROR MODEL TYPE IN
C ************ VARIANCE CALCULATIONS
C
C          MT  - MEASUREMENT TYPE
C                .EQ. 1 - RANDOM NVENTORY
C                .EQ. 2 - SYSTEMATIC INVENTORY
C                .EQ. 3 - RANDOM TRANSFER
C                .EQ. 4 - SYSTEMATIC TRANSFER
C          IB - BALANCE NUMBER
C          NC - COMPONENT NUMBER
C          NM - MEASUREMENT NUMBER
C          NS - SYSTEMATIC ERROR NUMBER
C          XX - COMPENSATING FACTOR
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
      COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
C
      GO TO (10,20,60,70), MT
C
   10 CONTINUE
C
C ************** RANDOM INVENTORY
C
      XX=XI(IB,NC)
      IF(XX.EQ.0)RETURN
      IF(IMTI(NC,NM).EQ.2)XX=XX/XIM(IB,NC,NM)
      RETURN
   20 CONTINUE
C
C ************** SYSTEMATIC INVENTORY
C
      XX=XI(IB,NC)
      IF(XX.EQ.0)RETURN
      ITST=IMTI(NC,NM)-2
      IF(ITST)30,50,40
   30 RETURN
   40 IF(NS.EQ.1)RETURN
   50 XX=XX/XIM(IB,NC,NM)
      RETURN
```

Fig. D-1. (cont)

```
   60 CONTINUE
C
C *************** RANDOM TRANSFER
C
      XX=T(IB,NC)
      IF(XX.EQ.0)RETURN
      IF(IMTT(NC,NM).EQ.2)XX=XX/TM(IB,NC,NM)
      RETURN
   70 CONTINUE
C
C *************** SYSTEMATIC TRANSFER
C
      XX=T(IB,NC)
      IF(XX.EQ.0)RETURN
      ITST=IMTT(NC,NM)-2
      IF(ITST)80,100,90
   80 RETURN
   90 IF(NS.EQ.1)RETURN
  100 XX=XX/TM(IB,NC,NM)
      RETURN
      END
      SUBROUTINE OUTDEC
C
C ***********************************************************************
C           WRITES OUTPUT FILES TO DECANAL CODE FOR TOTAL COLLAPSED CASE
C ***********************************************************************
C
C ***
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      INTEGER*2 DCNLIN, PVARA
      COMMON /TITLE/ ITITLE(40),IPVTI(30,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
C
C ************* WRITE A SCALAR EQUIVALENT OUTPUT TO DECANAL
C
      VDUM=0.
      DO 10 I=1,NBALP1
      IP1=I+1
      WRITE(NDECIN,20)XIT(I),S2IR(I),TTSUM(IP1),S2TRSM(IP1),VDUM
      WRITE(NDECIN,20)S2IB(I),CVI(I),VDUM
      WRITE(NDECIN,20)S2TBSM(IP1),CVT(IP1),VDUM
      WRITE(NDECIN,20)VDUM,VDUM
   10 CONTINUE
   20 FORMAT(8G15.7)
   30 RETURN
      END
```

Fig. D-1. (cont)

```
      SUBROUTINE PROCES(IPV,ITRIN)
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DIMENSION IPV(3)
C
C *************** CALLS THE INDIVIDUAL PROCESS ROUTINES TO
C *************** COMPUTE MEASURED VALUES
C
      IPROCI=ITIN(ITRIN)
      IF(IPROCI.EQ.0)GO TO 80
      GO TO(10,20,30,30,40,50,60,70),IPROCI
   10 CONTINUE
C
C *************** INVENTORY WITH ONE COMPONENT
C
      CALL INV1(IPV)
      RETURN
   20 CONTINUE
C
C *************** INVENTORY WITH TWO COMPONENTS
C
      CALL INV2(IPV)
      RETURN
   30 CONTINUE
C
C *************** COLUMN INVENTORY
C
      IC=NCOL(ITRIN)
      IT=IPROCI-2
      CALL COLUMN(IPV,IC,IT)
      RETURN
   40 CONTINUE
C
C *************** BATCH TRANSFER WITH ONE COMPONENT
C
      CALL TRAN1(IPV,ITRIN)
      RETURN
   50 CONTINUE
C
C *************** BATCH TRANSFER WITH TWO COMPONENTS
C
      CALL TRAN2(IPV,ITRIN)
      RETURN
   60 CONTINUE
C
C *************** TRANSFER AS A PRODUCT OF FLOW RATE AND CONCENTRATION
C
      CALL TRAN3(IPV,ITRIN)
      RETURN
   70 CONTINUE
      CALL TRAN4(IPV,ITRIN)
   80 RETURN
```

Fig. D-1. (cont)

```
      END
      SUBROUTINE PRTBAL
C
C *********************************************************************
C
C           COMPUTES MATERIAL BALANCES AND PRINTS OUT INVENTORIES,
C           TRANSFERS, AND THEIR ASSOCIATED VAIANCES ALONG WITH
C           MATERIAL BALANCES.
C
C
C *********************************************************************
C
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
C     DOUBLE PRECISION SUMMB,SUMMBS,DXMB,S2IS,S2TS,S2XMBS
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
      DIMENSION XMB(NBMXP1),S2XMB(NBMXP1)
C
C     TRACE 300
C
C *************** SUM INDIVIDUAL INVENTORY AND TRANSFER COMPONENTS
C
      DO 10 I=1,NBALP1
      XIT(I)=0.
      DO 10 J=1,NCI
      XIT(I)=XIT(I)+XI(I,J)
   10 CONTINUE
      DO 20 I=1,NTRN
      TT(I)=0.
      DO 20 J=1,NCT
      TT(I)=TT(I)+T(I,J)
   20 CONTINUE
C
   30 FORMAT(/'   I         XI            S2I          CVI          T
     1   S2T          CVT        MAT. BAL.    S2XMB         CUSUM        S2CUS
     1UM')
C
C *********************************************************************
C                       COMPUTE CUSUM
C *********************************************************************
C
      CALL CUSUM
   40 CONTINUE
      IF(IRUN.GT.1) GO TO 110
      IF(IRNSCH.GT.1)GO TO 110
C
C *************** COMPUTE MATERIAL BALANCES
```

Fig. D-1. (cont)

```
      C
            XMB(1)=0.0E0
            S2XMB(1)=0.
            IT=1
            DO 50 I=2,NBALP1
            XMB(I)=XIT(I-1)-XIT(I)+TTSUM(I)
            S2XMB(I)=S2I(I-1)+S2I(I)-2.*CVI(I-1)+S2TRSM(I)+S2TBSM(I)
         50 CONTINUE
            WRITE(NPROUT,60)
            IF(ISPNTI.EQ.0)WRITE(NPROUT,70)
            IF(ISPNTI.NE.0)WRITE(NPROUT,80) ISPNTI
            WRITE(NPROUT,90)
         60 FORMAT(1H1,' ***************************************************')
         70 FORMAT(/'  SUMMARY FOR ALL INVENTORIES AND TRANSFERS')
         80 FORMAT(/'  SUMMARY FOR INVENTORY AND TRANSFER NUMBER ='I2)
         90 FORMAT(/'  **************************************************')
            WRITE(NPROUT,30)
            DO 100 I=1,NBALP1
            S2SUM=S2TRSM(I)+S2TBSM(I)
            IM1=I-1
        100 WRITE(NPROUT,120) IM1,XIT(I),S2I(I),CVI(I),TTSUM(I),S2SUM,CVT(I),
           + XMB(I),S2XMB(I),CS(I),S2CS(I)
      C     WRITE(NPROUT,90)NBAL
      C  90 FORMAT(1H1' COMPARISON OF SAMPLED AND PROPAGATED VARIANCES OVER TH
      C    1E'I4' MATERIALS BALANCES'/'    (SYSTEMATIC ERRORS SHOULD BE SET TO
      C    1 ZERO FOR BEST COMPARISONS)')
      C     WRITE(NPROUT,100)
      C 100 FORMAT(/6X'INVENTORY')
      C     WRITE(NPROUT,110)S2IS,S2IA
      C 110 FORMAT(11X'SAMPLED VARIANCE ='1PD14.6,/
      C    1 11X'AVERAGE PROPAGATED VARIANCE ='1PE14.6)
      C     WRITE(NPROUT,120)
      C 120 FORMAT(/6X'TRANSFER',
      C     WRITE(NPROUT,110)S2TS,S2TA
      C     WRITE(NPROUT,125)
      C 125 FORMAT(/6X'MATERIALS BALANCE')
      C     WRITE(NPROUT,110)S2XMBS,S2XMBA
        110 CONTINUE
        120 FORMAT(1X,I4,2X,10(1PE12.4))
            RETURN
            END
            SUBROUTINE READEM(IPV)
      C
      C *******************************************************************
      C                  READ UNIT PROCESS INPUT DATA
      C *******************************************************************
      C
            PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
           1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
           2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
           2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
           4 NCOLMX=2)
      C
            COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
           + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
           + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
            COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
           $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
           1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
           2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
           3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
           4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
           5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      C
            INTEGER*2 DCNLIN, PVARA
```

Fig. D-1. (cont)

```
      COMMON /TITLE/ ITITLE(40),IPVTI(30,NPVMX),DCNLIN(3),MESINP(3),
     + PVARA(3)
C
      READ(NDAT,10)(IPVTI(I,IPV),I=1,30)
   10 FORMAT(30A2)
      READ(NDAT,*) SIGMAE(IPV),SIGMAN(IPV,1),SIGMAN(IPV,2),MESTYP(IPV),
     + INTCAL(IPV,1),INTCAL(IPV,2)
      IF(IMESPR.EQ.0)RETURN
      WRITE(NPROUT,20)IPV
   20 FORMAT(/'MEASUREMENT DATA FOR VARIABLE NO.'I3)
      WRITE(NPROUT,30)SIGMAE(IPV),SIGMAN(IPV,1),SIGMAN(IPV,2),
     + MESTYP(IPV),INTCAL(IPV,1),INTCAL(IPV,2)
   30 FORMAT(10X,3F10.5,3I7)
      RETURN
      END
      SUBROUTINE REDPV
C
C                READS PROCESS VARIABLE (PVI AND PVT) ARRAYS
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /PVCOMS/ PVIS(NBMXP3,NPVIMX),PVTS(NTMXP2,NPVTMX)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NC1,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      READ(NPVIN) N
C
C *************** CHECK IF N EQUALS NPV
C
      IF(N.EQ.NPV)GO TO 20
      WRITE(1,10) N, NPV
      WRITE(NPROUT,10) N, NPV
   10 FORMAT(//' NUMBER OF DIFFERENT PROCESS VARIABLES IN PV ARRAY ='I3
     1 /' INPUT DATA SETS THIS NUMBER (NPV) AT 'I3
     2 /' RUN TERMINATED BECAUSE OF THIS INCONSISTENCY')
      CALL CLOSEM
   20 CONTINUE
      WRITE(NPROUT,30) N
   30 FORMAT(/5X'NUMBER OF DIFFERENT PROCESS VARIABLES IN ARRAY ='I3)
      READ(NPVIN)(IPVTRN(I),I=1,N)
      WRITE(NPROUT,40)(IPVTRN(I),I=1,N)
   40 FORMAT(5X'IPVTRN - TRANSFER INDICATOR ARRAY FOR PROCESS VARIABLES
     1(1 FOR TRANSFER)'/(5X,60I2))
      READ(NPVIN)(NPVCT(I),I=1,N)
      WRITE(NPROUT,50)(NPVCT(I),I=1,N)
   50 FORMAT(5X'NUMBER OF VARIABLES IN PV ARRAY FOR EACH PROCESS VARIABL
     1E' /(5X,30I4))
C
C *************** CHECK TO SEE IF DIMENSIONS OF PVI AND PVT ARRAYS
C *************** ARE BEING EXCEEDED
C
      DO 110 I=1,N
      IF(IPVTRN(I).GT.0)GO TO 70
      IF(NPVCT(I).LE.NBMXP3)GO TO 110
```

Fig. D-1. (cont)

```
      WRITE(1,6Ø)NPVCT(I)
   6Ø FORMAT(//' ATTEMPTED TO READ IN 'I3' VALUES INTO PVI ARRAY')
      GO TO 9Ø
   7Ø IF(NPVCT(I).LE.NTMXP2)GO TO 11Ø
      WRITE(1,8Ø)NPVCT(I)
      WRITE(NPROUT,8Ø)NPVCT(I)
   8Ø FORMAT(//' ATTEMPTED TO READ IN 'I3' VALUES INTO PVT ARRAY')
   9Ø WRITE(1,1ØØ)
  1ØØ FORMAT(/'*********** RUN TERMINATED ***********')
      CALL CLOSEM
  11Ø CONTINUE
C
C ************** SET NPVIT ARRAY
C
      KI=Ø
      KT=Ø
      DO 13Ø I=1,N
      IF(IPVTRN(I).GT.Ø)GO TO 12Ø
      KI=KI+1
      NPVIT(I)=KI
      GO TO 13Ø
  12Ø KT=KT+1
      NPVIT(I)=KT
  13Ø CONTINUE
C
C ************** INPUT VALUES TO THE PVI AND PVT ARRAYS
C
      DO 15Ø J=1,N
      NPVITJ=NPVIT(J)
      NFC=NPVCT(J)
      IF(IPVTRN(J).GT.Ø)GO TO 14Ø
      READ(NPVIN) (PVIS(I,NPVITJ),I=1,NFC)
      GO TO 15Ø
  14Ø READ(NPVIN) (PVTS(I,NPVITJ),I=1,NFC)
  15Ø CONTINUE
      RETURN
      END
      SUBROUTINE RNORM
C
      COMMON /MSRNRM/ RN1,RN2,ISTRM
C
C     DATA TWOPI /6.2831853Ø7/
      PARAMETER (TWOPI=6.2831853Ø7)
C
C **********************************************************************
C               RETURNS TWO NORMALLY DISTRIBUTED RANDOM NUMBERS
C **********************************************************************
C
      U1=DRAND(ISTRM)
      U2=DRAND(ISTRM)
      IF(U1.LE.Ø)U1=1.E-6
      T=SQRT(-2*ALOG(U1))
      RN1=T*COS(TWOPI*U2)
      RN2=T*SIN(TWOPI*U2)
      RETURN
      END
      SUBROUTINE SETMAS
C
C ************ COMPUTES THE INPUT VARIANCE AND CORRELATION INDICATOR
C                ARRAYS FOR EACH TRANSFER AND INVENTORY
C
C ************ KEYS ON PROCESS NUMBER TO SET UP THE DESIRED ARRAYS
C
C **** PROCESS NUMBER INDEX
C           1 - INVENTORY AS SINGLE VARIABLE
```

Fig. D-1. (cont)

```
C            2 - INVENTORY AS PRODUCT OF TWO VARIABLES
C            3 - INVENTORY FROM COLUMNS OF "A" OR "S" TYPE
C            4 - INVENTORY FROM COLUMNS OF "B" OR "Z" TYPE
C            5 - TRANSFER AS SINGLE VARIABLE
C            6 - TRANSFER AS PRODUCT OF TWO VARIABLES
C            7 - TRANSFER AS A PRODUCT OF CONCENTRATION AND FLOW RATE
C
C        EXPLANATION OF VARIABLES
C
C            ISPNTI - SPECIFIC UNIT "TRANSFER-INVENTORY" SET NUMBER (0 GIVES ALL)
C            ITIN   - TRANSFER INVENTORY TYPE INDICATOR
C            NCINOW - INVENTORY NUMBER COUNTER
C            NCTNOW - TRANSFER NUMBER COUNTER
C            NTRIN  - NUMBER OF TRANSFER OR INVENTORY TERMS
C
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      INTEGER TSCI,TCI
C
      COMMON /MSINCM/ ICI(NCIMX,NCIMX,NMIMX), ISCI(NCIMX,NMIMX),
     + ITCI(NCIMX,NCTMX,NCMX), TCI(NCTMX,NCTMX,NMTMX),TSCI(NCTMX,NMTMX),
     +  VIR(NCIMX,NMIMX), VTR(NCTMX,NMTMX),VIS(NCIMX,NMIMX,2),
     +  VTS(NCTMX,NMTMX,2), SIR(NCIMX,NMIMX),STR(NCTMX,NMTMX),
     +  SIS(NCIMX,NMIMX,2), STS(NCTMX,NMTMX,2)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     S NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      COMMON /MESPAR/ SIGMAE(NPVMX),SIGMAN(NPVMX,2),SIG2E(NPVMX),
     + SIG2N(NPVMX,2),MESTYP(NPVMX),INTCAL(NPVMX,2),ISTRPV(NPVMX),IZE,
     + IMTI(NCIMX,NMIMX), IMTT(NCTMX,NMTMX),ISTRNR(NPVMX),ISTRNS(NPVMX)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
C
      COMMON /RUNCOM/ NRUN, IRUN, ISPNTI
C
      DIMENSION IPVN(3), JPNTR(2,2)
C
      DATA JPNTR/1,1,2,3/
C
C *************** ZERO THE MASSAGE INPUT ARRAYS
C
      DO 20 I=1,NCIMX
      DO 20 J=1,NMIMX
      VIR(I,J)=0.
      SIR(I,J)=0.
      ISCI(I,J)=0
      DO 10 K=1,NSIMX
      VIS(I,J,K)=0.
      SIS(I,J,K)=0.
   10 CONTINUE
      DO 20 L=1,NCIMX
      ICI(I,L,J)=0
   20 CONTINUE
      DO 40 I=1,NCTMX
```

Fig. D-1. (cont)

```
      DO 40 J=1,NMTMX
      VTR(I,J)=0.
      STR(I,J)=0.
      TSCI(I,J)=0
      DO 30 K=1,NSTMX
      VTS(I,J,K)=0.
      STS(I,J,K)=0.
   30 CONTINUE
      DO 40 L=1,NCTMX
      TCI(I,L,J)=0
   40 CONTINUE
      DO 50 I=1,NCIMX
      DO 50 J=1,NCTMX
      DO 50 K=1,NCMX
      ITCI(I,J,K)=0
   50 CONTINUE
C
C ************ SET THE MASSAGE INPUT ARRAYS
C
C     TRACE 999
      NCINOW=0
      NCTNOW=0
      NPVCNT=0
      NPVI=0
      NPVT=0
      NCFRC=0
      NCGF=0
      NRNS=0
      ISTRT=1
      IFIN=NTRIN
      IF(ISPNTI.EQ.0)GO TO 60
      ISTRT=ISPNTI
      IFIN=ISTRT
   60 CONTINUE
      NCI=0
      NCT=0
      NMI=1
      NMT=1
      NSI=1
      NST=1
      NCOLUM=0
      DO 350 ITRIN=ISTRT,IFIN
      DO 70 I=1,3
      IPVN(I)=IPVNO(I,ITRIN)
   70 CONTINUE
      ITINN=ITIN(ITRIN)
      IF(ITINN.EQ.0)GO TO 350
      GO TO (80,80,110,110,170,170,190,240),ITINN
   80 CONTINUE
C
C ******** INVENTORIES
C
C                   VIR(NCI,NMI)     - VARIANCE OF INVENTORY RANDOM ERRORS
C                   VIS(NCI,NMI,NSI) - VARIANCE OF INVENTORY SYSTEMATIC ERRORS
C                   ICI(NCI,NCI,NMI) - INVENTORY CORRELATION INDICATOR
C                   ISCI(NCI,NMI)    - INVENTORY SEQUENTIAL CORRELATION INDICATOR
C
C
      NPVI=NPVI+ITINN
      NCI=NCI+1
      IF(ITINN.EQ.2)NMI=2
      NCINOW=NCINOW+1
      DO 100 I=1,ITINN
      NPVCNT=NPVCNT+1
      IPVNAR(NPVCNT)=IPVN(I)
```

Fig. D-1. (cont)

```
      NRNS=NRNS+1
      ISTRNR(IPVN(I))=NRNS
      VIR(NCINOW,I)=SIG2E(IPVN(I))
      SIR(NCINOW,I)=SQRT(VIR(NCINOW,I))
      IMTI(NCINOW,I)=MESTYP(IPVN(I))
      IF(NSI.EQ.0)GO TO 100
      DO 90 J=1,2
      VIS(NCINOW,I,J)=SIG2N(IPVN(I),J)
      SIS(NCINOW,I,J)=SQRT(VIS(NCINOW,I,J))
      IF(VIS(NCINOW,I,J).EQ.0)GO TO 90
      NRNS=NRNS+1
      IF(J.EQ.1)ISTRNS(IPVN(I))=NRNS
      IF(J.EQ.2)NSI=2
      ICI(NCINOW,NCINOW,I)=J
      ISCI(NCINOW,I)=J
   90 CONTINUE
  100 CONTINUE
      GO TO 150
  110 CONTINUE
C
C ************ COLUMN INVENTORIES
C
      NCOLUM=NCOLUM+1
      NPVI=NPVI+5
      DO 130 I=1,3
      NPVCNT=NPVCNT+1
      IPVNAR(NPVCNT)=IPVN(I)
      NCINOW=NCINOW+1
      NRNS=NRNS+1
      ISTRNR(IPVN(I))=NRNS
      VIR(NCINOW,1)=SIG2E(IPVN(I))
      SIR(NCINOW,1)=SQRT(VIR(NCINOW,1))
      IMTI(NCINOW,1)=MESTYP(IPVN(I))
      IF(NSI.EQ.0)GO TO 130
      DO 120 J=1,2
      VIS(NCINOW,1,J)=SIG2N(IPVN(I),J)
      SIS(NCINOW,1,J)=SQRT(VIS(NCINOW,1,J))
      IF(VIS(NCINOW,1,J).EQ.0)GO TO 120
      NRNS=NRNS+1
      IF(J.EQ.1)ISTRNS(IPVN(I))=NRNS
      IF(J.EQ.2)NSI=2
      ICI(NCINOW,NCINOW,1)=J
      ISCI(NCINOW,1)=J
  120 CONTINUE
  130 CONTINUE
      IPT4=IPVN(3)+1
      IPVTRN(IPT4)=-1
      IPVTRN(IPT4+1)=-1
      NCI=NCI+3
      IF(NCOLUM.GT.1)GO TO 140
      NCI=NCI+1
  140 CONTINUE
  150 CONTINUE
      IF(NCINOW.LE.NCI)GO TO 350
      WRITE(1,160)
      WRITE(NPROUT,160)
  160 FORMAT(//' RUN TERMINATED IN ROUTINE SETMAS WITH NCINOW .GT. NC
     1I')
      CALL CLOSEM
  170 CONTINUE
      ITMP=ITINN-4
      NPVT=NPVT+ITMP
      IF(ITMP.EQ.2)NMT=2
C
C ******** TRANSFERS
```

Fig. D-1. (cont)

```
C
C                 VTR(NCT,NMT)       - VARIANCE OF TRANSFER RANDOM ERRORS
C                 VTS(NCT,NMT,NST)   - VARIANCE OF TRANSFER SYSTEMATIC ERRORS
C                 TCI(NCT,NCT,NMT)   - TRANSFER CORRELATION INDICATOR
C                 TSCI(NCT,NMT)      - TRANSFER SEQUENTIAL CORRELATION INDICATOR
C
      NCTNOW=NCTNOW+1
      IF=ITINN-4
      NCT=NCT+1
      DO 180 I=1,IF
      NPVCNT=NPVCNT+1
      IPVNAR(NPVCNT)=IPVN(I)
      NRNS=NRNS+1
      ISTRNR(IPVN(I))=NRNS
      VTR(NCTNOW,I)=SIG2E(IPVN(I))
      STR(NCTNOW,I)=SQRT(VTR(NCTNOW,I))
      IMTT(NCTNOW,I)=MESTYP(IPVN(I))
      DO 180 J=1,2
      NCAL(NCTNOW,I,J)=INTCAL(IPVN(I),J)
      VTS(NCTNOW,I,J)=SIG2N(IPVN(I),J)
      STS(NCTNOW,I,J)=SQRT(VTS(NCTNOW,I,J))
      IF(VTS(NCTNOW,I,J).EQ.0)GO TO 180
      NRNS=NRNS+1
      IF(J.EQ.1)ISTRNS(IPVN(I))=NRNS
      IF(J.EQ.2)NST=2
      TCI(NCTNOW,NCTNOW,I)=J
      TSCI(NCTNOW,I)=J
  180 CONTINUE
      GO TO 330
  190 CONTINUE
C
C ************** TRANSFER AS A PRODUCT OF FLOW RATE AND CONCENTRATION
C
C    T = (C(I-1)*F(I-1) + C(I)*F(I) + (C(I-1)*F(I) + C(I)*F(I-1)/2.))*DT/3.
C
      NPVT=NPVT+2
      NCT=NCT+4
      NMT=2
      IS=NCTNOW+1
      DO 210 I=1,4
      NCTNOW=NCTNOW+1
      DO 210 J=1,NMT
      IF(I.NE.1)GO TO 200
      NPVCNT=NPVCNT+1
      IPVNAR(NPVCNT)=IPVN(J)
      NRNS=NRNS+1
      ISTRNR(IPVN(J))=NRNS
  200 CONTINUE
      VTR(NCTNOW,J)=SIG2E(IPVN(J))
      STR(NCTNOW,J)=SQRT(VTR(NCTNOW,J))
      IMTT(NCTNOW,J)=MESTYP(IPVN(J))
      DO 210 K=1,2
      NCAL(NCTNOW,J,K)=INTCAL(IPVN(J),K)
      VTS(NCTNOW,J,K)=SIG2N(IPVN(J),K)
      STS(NCTNOW,J,K)=SQRT(VTS(NCTNOW,J,K))
      IF(I.NE.1)GO TO 210
      IF(VTS(NCTNOW,J,K).EQ.0)GO TO 210
      NRNS=NRNS+1
      IF(K.EQ.1)ISTRNS(IPVN(J))=NRNS
      IF(K.EQ.2)NST=2
  210 CONTINUE
C
C ************** TRANSFER CORRELATION INDICATORS
C
      IF=IS+3
```

Fig. D-1. (cont)

```
      DO 230 I=1,2
      DO 230 J=1,NST
      IF(VTS(IS,I,J).EQ.0)GO TO 230
      DO 220 II=IS,IF
      TSCI(II,I)=J
      DO 220 JJ=IS,IF
      TCI(II,JJ,I)=J
  220 CONTINUE
  230 CONTINUE
      NCFRC=NCFRC+1
      ICFR(NCFRC)=IS
      GO TO 330
  240 CONTINUE
C
C ****** TRANSFER AS A PRODUCT OF CONCENTRATION AND WEIGHT CHANGE
C
C                 T = G * ( FI - FF )
C
C
      NPVT=NPVT+3
      NCT=NCT+2
      NMT=2
      IS=NCTNOW+1
      DO 300 I=1,2
      NCTNOW=NCTNOW+1
      DO 300 J=1,NMT
      IF(I.NE.2) GO TO 250
      IF(J.EQ.1)GO TO 300
  250 JJ=JPNTR(I,J)
      NPVCNT=NPVCNT+1
      IPVNAR(NPVCNT)=IPVN(JJ)
      NRNS=NRNS+1
      ISTRNR(IPVN(JJ))=NRNS
      VTR(NCTNOW,J)=SIG2E(IPVN(JJ))
      STR(NCTNOW,J)=SQRT(VTR(NCTNOW,J))
      IMTT(NCTNOW,J)=MESTYP(IPVN(JJ))
      DO 300 K=1,2
      NCAL(NCTNOW,J,K)=INTCAL(IPVN(JJ),K)
      VTS(NCTNOW,J,K)=SIG2N(IPVN(JJ),K)
      STS(NCTNOW,J,K)=SQRT(VTS(NCTNOW,J,K))
      IF(VTS(NCTNOW,J,K).EQ.0)GO TO 300
      IF(K.EQ.2)GO TO 290
      IF(J.EQ.1)GO TO 270
      IF(I.EQ.2)GO TO 260
      NRNS=NRNS+1
      NRNSAV=NRNS
      GO TO 280
  260 NRNSMX=NRNS
      NRNS=NRNSAV
      GO TO 280
  270 NRNS=NRNS+1
  280 ISTRNS(IPVN(JJ))=NRNS
      GO TO 300
  290 NRNS=NRNS+1
      NST=2
  300 CONTINUE
      NRNS=NRNSMX
C
C ************** TRANSFER CORRELATION INDICATORS
C
      IF=IS+1
      DO 320 I=1,2
      DO 320 J=1,NST
      IF(VTS(IS,I,J).EQ.0)GO TO 320
      DO 310 II=IS,IF
```

Fig. D-1. (cont)

```
      TSCI(II,I)=J
      DO 31Ø JJ=IS,IF
      TCI(II,JJ,I)=J
  31Ø CONTINUE
  32Ø CONTINUE
      NCGF=NCGF+1
      ICGF(NCGF)=IS
  33Ø CONTINUE
      IF(NCTNOW.LE.NCT)GO TO 35Ø
      WRITE(1,34Ø)
      WRITE(NPROUT,34Ø)
  34Ø FORMAT(//' RUN TERMINATED IN ROUTINE SETMAS WITH NCTNOW .GT. NC
     1T')
      CALL CLOSEM
  35Ø CONTINUE
C
C *************** CHECK FOR INVENTORY-TRANSFER CORRELATIONS
C
      INTRNC=Ø
      DO 36Ø I=1,NCI
      DO 36Ø J=1,NCT
      DO 36Ø K=1,NC
      IF(ITCI(I,J,K).GT.Ø)INTRNC=1
  36Ø CONTINUE
      NNSTRM=NRNS
C
C *************** CHECK FOR OVERFLOW OF PVI AND PVT ARRAYS
C
      IF(NPVI.LE.NPVIMX) GO TO 38Ø
      WRITE(NPROUT,37Ø)
      WRITE(1,37Ø)
  37Ø FORMAT(' RUN TERMINATED WITH NPVI GREATER THAN NPVIMX')
      CALL CLOSEM
  38Ø IF(NPVT.LE.NPVTMX) GO TO 4ØØ
      WRITE(NPROUT,39Ø)
      WRITE(1,39Ø)
  39Ø FORMAT(' RUN TERMINATED WITH NPVT GREATER THAN NPVTMX')
      CALL CLOSEM
  4ØØ CONTINUE
      IF(MASPRT.EQ.Ø)RETURN
      WRITE(NPROUT,41Ø)
  41Ø FORMAT(///' VARIANCE OF INVENTORY RANDOM ERROR - VIR(NCI,NMI)')
      WRITE(NPROUT,42Ø)
  42Ø FORMAT(/5X' COMPONENT NO.    MEASUREMENT NO.     VARIANCE')
      CALL WRTR(VIR,NCIMX,NCI,NMI,NPROUT)
C
      WRITE(NPROUT,43Ø)
  43Ø FORMAT(///'VARIANCE OF TRANSFER RANDOM ERROR - VTR(NCT,NMT)')
      WRITE(NPROUT,42Ø)
      CALL WRTR(VTR,NCTMX,NCT,NMT,NPROUT)
C
      WRITE(NPROUT,44Ø)
  44Ø FORMAT(///' VARIANCE OF INVENTORY BIAS ERROR - VIS(NCI,NMI,NSI)')
      WRITE(NPROUT,45Ø)
  45Ø FORMAT(/5X' COMPONENT NO.   MEASUREMENT NO.   BIAS NO.  VARIANCE')
      CALL WRTB(VIS,NCIMX,NMIMX,NCI,NMI,NSI,NPROUT)
C
      WRITE(NPROUT,46Ø)
  46Ø FORMAT(///' VARIANCE OF TRANSFER BIAS ERROR - VTS(NCT,NMT,NST)')
      WRITE(NPROUT,45Ø)
      CALL WRTB(VTS,NCTMX,NMTMX,NCT,NMT,NST,NPROUT)
C
      WRITE(NPROUT,47Ø)
  47Ø FORMAT(///' INVENTORY CORRELATION INDICATOR - ICI(NCI,NCI,NMI)')
      CALL WRTC(ICI,NCIMX,NCIMX,NCI,NCI,NMI,NPROUT)
```

Fig. D-1. (cont)

```
C
      WRITE(NPROUT,480)
  480 FORMAT(///' TRANSFER CORRELATION INDICATOR - TCI(NCT,NCT,NMT)')
      CALL WRTC(TCI,NCTMX,NCTMX,NCT,NCT,NMT,NPROUT)
C
      WRITE(NPROUT,490)
  490 FORMAT(///' INVENTORY-TRANSFER CORRELATION INDICATOR - ITCI(NCI,NC
     1T,NC)')
      CALL WRTC(ITCI,NCIMX,NCTMX,NCI,NCT,NC,NPROUT)
C
      WRITE(NPROUT,500)
  500 FORMAT(///' INVENTORY SEQUENTIAL CORRELATION INDICATOR - ISCI(NCI,
     1NMI)')
      CALL WRTS(ISCI,NCIMX,NCI,NMI,NPROUT)
C
      WRITE(NPROUT,510)
  510 FORMAT(///' TRANSFER SEQUENTIAL CORRELATION INDICATOR - TSCI(NCT,N
     1MT)')
      CALL WRTS(TSCI,NCTMX,NCT,NMT,NPROUT)
C
      RETURN
      END
      SUBROUTINE STNDEV
C
C ***********************************************************************
C            COMPUTES SAMPLE STANDARD DEVIATIONS FOR MULTIPLE RUNS
C ***********************************************************************
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      PARAMETER (ZP=1.96)
C
      DOUBLE PRECISION SUM,SUMSQ,XNRUN
C
      COMMON /SAMPLE/ SUM(NBMXP1), SUMSQ(NBMXP1)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
      COMMON /RUNCOM/ NRUN,IRUN,ISPNTI
      DIMENSION IWARN(14,2)
      DATA IWARN/'OK
     1  '***** RATIO OUTSIDE INTERVAL'/
C
      WRITE(NPROUT,10) NRUN
   10 FORMAT(1H1,'*******************************************************'
     1 //'  RESULTS OF MONTE CARLO SIMULATION WITH 'I5' SAMPLES'//
     2'*******************************************************')
C
C ***********************************************************************
C                         COMPUTE SAMPLE VARIANCE
C ***********************************************************************
```

Fig. D-1. (cont)

```
C
      SUMSQ(1)=0.
      XNRUN=DBLE(FLOAT(NRUN))
      TMP1=2.*(NRUN-1)
      TMP2=SQRT(TMP1-1.)
      X2UP=(ZP+TMP2)**2/TMP1
      X2LOW=(-ZP+TMP2)**2/TMP1
      WRITE(NPROUT,20) X2UP, X2LOW
   20 FORMAT(/' CHI-SQUARE/(N-1) RATIO FOR 95% CONFIDENCE'/
     1 5X'UPPER LIMIT ='G14.6/5X'LOWER LIMIT ='G14.6)
      DO 30 I=2,NBALP1
      SUMSQ(I)=(SUMSQ(I)-SUM(I)**2/XNRUN)/(XNRUN-1.D0)
      SUM(I)=SUM(I)/XNRUN
   30 CONTINUE
      WRITE(NPROUT,40)
   40 FORMAT(/16X'CUSUM'12X'CUSUM'11X'CUSUM'11X'RATIO'/
     1 3X'BALANCE'6X'SAMPLE'9X'PROPAGATED'8X'SAMPLE'10X'SAMPLE/ '
     2 /3X'NUMBER'6X'AVERAGE'10X'VARIANCE'8X'VARIANCE'
     3 7X'PROPAGATED'//)
      NOUT=0
      DO 70 I=2,NBALP1
      SMPVAR=SUMSQ(I)
      PRPVAR=S2CS(I)
      RATIO=SMPVAR/PRPVAR
      IVIOL=1
      IM1=I-1
      IF(RATIO.LT.X2LOW)GO TO 50
      IF(RATIO.LT.X2UP)GO TO 60
   50 IVIOL=2
   60 CONTINUE
      WRITE(NPROUT,80)IM1,SUM(I),PRPVAR,SMPVAR,RATIO,(IWARN(J,IVIOL),
     + J=1,14)
   70 CONTINUE
   80 FORMAT(3X,I4,3X,1PD14.6,3X,G14.6,3X,G14.6,3X,G14.6,
     1 5X,14A2)
      RETURN
      END
      SUBROUTINE TRAN1(IPV,ITRIN)
C
C************** MODELS TRANSFER WITH ONE MEASUREMENT
C
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     S NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     I NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DIMENSION IPV(1)
      ICTRN=ICTRN+1
```

Fig. D-1. (cont)

```
      IF(ICTRN.LE.NCT)GO TO 2Ø
      WRITE(1,1Ø) IPV(1)
      WRITE(NPROUT,1Ø) IPV(1)
   1Ø FORMAT(//' RUN TERMINATED IN SUBROUTINE TRAN1 WITH ICTRN .GT. NCT'
     1 /'    PROCESS VARIABLE NUMBER ='I3)
      CALL CLOSEM
   2Ø CONTINUE
      IPVTI=NPVIT(IPV(1))
      DO 3Ø I=1,NTRNP2
      TM(I,ICTRN,1)=PVT(I+1,IPVTI)
   3Ø CONTINUE
C
C************** TRANSFER
C
      DO 4Ø I=1,NTRNP1
      TM(I,ICTRN,1)=ITRAN(ITRIN)*TM(I,ICTRN,1)
      T(I,ICTRN)=TM(I,ICTRN,1)
   4Ø CONTINUE
      IF(NMT.EQ.1)GO TO 6Ø
      DO 5Ø I=1,NTRNP1
      TM(I,ICTRN,2)=1.
   5Ø CONTINUE
   6Ø RETURN
      END
      SUBROUTINE TRAN2(IPV,ITRIN)
C
C************** BATCH TRANSFER MODEL WITH TWO MEASUREMENTS
C
C
C
      PARAMETER (NBALMX=1Ø5, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=1Ø, NPVTMX=8, NCIMX=1Ø, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=2Ø,
     4 NCOLMX=2)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXPI), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
      DIMENSION IPV(1)
C
C
C               PV(1,IPV) = PROCESS VALUE OF TRANSFER VOLUME
C               PV(2,IPV) = PROCESS VALUE OF CONCENTRATION
C
C
C     TRACE 1ØØØ
C
      ICTRN=ICTRN+1
      IF(ICTRN.LE.NCT)GO TO 2Ø
      WRITE(1,1Ø) IPV(1)
      WRITE(NPROUT,1Ø) IPV(1)
   1Ø FORMAT(//' RUN TERMINATED IN SUBROUTINE TRAN2 WITH ICTRN .GT. NCT'
```

Fig. D-1. (cont)

```
     1 /'    PROCESS VARIABLE NUMBER ='I3)
      CALL CLOSEM
   20 CONTINUE
      IPVTI1=NPVIT(IPV(1))
      IPVTI2=NPVIT(IPV(2))
      DO 30 I=1,NTRNP2
      IP1=I+1
      TM(I,ICTRN,1)=PVT(IP1,IPVTI1)
      TM(I,ICTRN,2)=PVT(IP1,IPVTI2)
   30 CONTINUE
C
C*************** TRANSFER
C
      DO 40 I=1,NTRNP1
      TM(I,ICTRN,1)=TM(I,ICTRN,1)*ITRAN(ITRIN)
      T(I,ICTRN)=TM(I,ICTRN,1)*TM(I,ICTRN,2)
   40 CONTINUE
   50 RETURN
      END
      SUBROUTINE TRAN3(IPV,ITRIN)
C
C *************** MODELS TRANSFERS WHICH ARE A PRODUCT OF A
C *************** CONCENTRATION AND FLOW RATE, IE
C
C     T = (C(I-1)*F(I-1) + C(I)*F(I) + (C(I-1)*F(I) + C(I)*F(I-1)/2.))*DT/3.
C
C ******* IN THIS CASE THE TRANSFER HAS FOUR COMPONENTS WITH 2
C ******* MEASUREMENTS PER COMPONENT.
C
C           PV(N,1) = CONCENTRATION
C           PV(N,2) = FLOW RATE
C           C       = CONCENTRATION
C           F       = FLOW RATE
C           IPV     = PROCESS VARIABLE NUMBER
C           ITRIN   = TRANSFER INVENTORY NUMBER
C           DT      = TIME INTERVAL BETWEEN MEASUREMENTS
C
      PARAMETER (NBALMX=105, NBMXP1=NBALMX+1, NBMXP2=NBALMX+2,
     1 NBMXP3=NBALMX+3, NTRNMX=515, NTMXP1=NTRNMX+1, NTMXP2=NTRNMX+2,
     2 NPVMX=12, NPVIMX=10, NPVTMX=8, NCIMX=10, NCTMX=8, NMIMX=2,
     2 NMTMX=2, NSIMX=2, NSTMX=2, NCMX=2, MXPMX=2, NSTRMX=20,
     4 NCOLMX=2)
C
      COMMON /PVCOM/ PVI(NBMXP3,NPVIMX),PVT(NTMXP2,NPVTMX),IPVTRN(NPVMX)
C
      COMMON /VAR/ XI(NBMXP2,NCIMX), S2I(NBMXP1), CVI(NBMXP1),T(NTMXP1,
     + NCTMX), XIM(NBMXP2,NCIMX,NMIMX), TM(NTMXP1,NCTMX,NMTMX),
     + S2T(NTRNMX), CVT(NTRNMX), CS(NBMXP1), S2CS(NBMXP1), TT(NTRNMX),
     + XIT(NBMXP1), S2IR(NBMXP1), S2IB(NBMXP1), S2TR(NTRNMX),
     + S2TB(NTRNMX), TTSUM(NBMXP1), S2TRSM(NBMXP1), S2TBSM(NBMXP1)
C
C
      COMMON /CON/ NBAL,NBALP1,NBALP2,NTRPBL,NTRN,NTRNP1,NTRNP2,
     $ NPROUT,DT,NCAL(NCTMX,NMTMX,2),NPVIT(NPVMX),IBLANK,
     1 NPVIN,NDECIN,IPRPV,NDAT,IRNSCH,ITIN(NPVMX),NCITST,NFSEED,INTRNC,
     2 ITRAN(NPVMX),IPVNO(5,NPVMX),NTRIN,MASPRT,ITIPRP,IMESPR,ICLAPS,
     3 ICTRN,ICINV,NCOL(NPVMX),HCC(5,NCOLMX),VCC(2,NCOLMX),CCC(3,NCOLMX)
     4 ,NPV,NCI,NCT,NMI,NMT,NSI,NST,NC,MXP,NNSTRM,NCOLUM,NCFRC,ICFR(4),
     5 NPVCT(NPVMX),NPVI,NPVT,NCGF,ICGF(4),NPVCNT,IPVNAR(NPVMX)
C
C
      DIMENSION C(NTMXP2),F(NTMXP2),IPV(1)
C
      COMMON /WORK/ X(NTMXP2,3)
C
```

Fig. D-1. (cont)

```
      EQUIVALENCE (X(1,1) , C) , (X(1,2) , F)
C
C
C ************** DETERMINE MEASURED VALUES FOR CONCENTRATIONS AND
C ************** FLOW RATES
C
      NBALP3=NBALP2+1
      IPVTI1=NPVIT(IPV(1))
      IPVTI2=NPVIT(IPV(2))
      DO 10 I=1,NTRNP2
      IP1=I+1
      C(I)=PVT(IP1,IPVTI1)
      F(I)=PVT(IP1,IPVTI2)
   10 CONTINUE
C
C *************** MULTIPLY THE CONCENTRATIONS BY A CONSTANT
C
      CON=ITRAN(ITRIN)*DT/3.
      DO 20 I=1,NTRNP2
      C(I)=C(I)*CON
   20 CONTINUE
C
C *************** FORM THE 8 TRANSFER MEASUREMENTS
C
      IC1=ICTRN+1
      IC2=ICTRN+2
      IC3=ICTRN+3
      IC4=ICTRN+4
      ICTRN=IC4
      IF(ICTRN.LE.NCT)GO TO 40
      WRITE(1,30) IPV
      WRITE(NPROUT,30) IPV
   30 FORMAT(//' RUN TERMINATED IN SUBROUTINE TRAN3 WITH ICTRN .GT. NCT'
     1 /'    PROCESS VARIABLE NUMBER ='I3)
      CALL CLOSEM
   40 CONTINUE
      DO 50 I=1,NTRNP1
      IP1=I+1
      TM(I,IC1,1)=C(I)
      TM(I,IC1,2)=F(I)
      TM(I,IC2,1)=C(IP1)
      TM(I,IC2,2)=F(IP1)
      TM(I,IC3,1)=C(I)/2
      TM(I,IC3,2)=F(IP1)
      TM(I,IC4,1)=C(IP1)/2
      TM(I,IC4,2)=F(I)
   50 CONTINUE
C
C *************** FORM THE FOUR TRANSFER COMPONENTS
C
      DO 60 I=1,NTRNP1
      DO 60 J=IC1,IC4
      T(I,J)=TM(I,J,1)*TM(I,J,2)
   60 CONTINUE
C
      RETURN
      END
      SUBROUTINE TRAN4(IPV,ITRIN)
C
C *********** MODELS THE PRODUCT OF CONCENTRATION AND THE DIFFERENCE
C ***********   IN WEIGHTS.
C
C            T = G * ( FI - FF)
C
C                WHERE:
```

Fig. D-1. (cont)

REFERENCES

1. E. A. Kern and D. P. Martinez, "Users Manual for a Process Model Code," Los Alamos National Laboratory report LA-8761-M (March 1981).

2. J. T. Markin, A. L. Baker, and J. P. Shipley, "Implementing Advanced Data Analysis Techniques in Near-Real-Time Materials Accounting," Nucl. Mater. Manage. IX, 236-244 (1980).

3. A. Papoulis, Probability, Random Variables, and Stochastic Processes (McGraw-Hill Book Company, Inc., New York, 1965).

4. M. R. Spiegel, Probability and Statistics, Schaum's Outline Series in Mathematics (McGraw-Hill Book Company, Inc., New York, 1975).

5. W. J. Dixon and F. J. Massey, Jr., Introduction to Statistical Analysis (McGraw-Hill Book Company, Inc., New York, 1969).

★U.S. GOVERNMENT PRINTING OFFICE: 1982-0-576-020/119